USB flows in the Great River: Classic Tradecraft is still alive

Hiroshi Takeuchi
Security Research Center

VB2023 London

Co.Tomorrowing

MACNICA

46% - 82%

Publisher	Report	Percentage of Exploiting public-facing device	
SecureWorks	2022 State of the Threat: A Year in Review	52%	Exploitation of remote services 52%
COVEWARE	Fewer Ransomware Victims Pay, as Median Ransom Falls in Q2 2022	50%	RDP Compromise approx. 30% Software Vulnerability approx. 20%+
Palo Alto	Attackers Move Quickly to Exploit High-Profile Zero Days: Insights From the 2022 Unit 42 Incident Response Report	46%	Software vulnerabilities 31% Brute force credential attacks 9% Previously leaked credentials 6%
SOPHOS	The Active Adversary Playbook 2022	55%	Exploited Vulnerability 47% Compromised Credentials 5% Brute Force Attack 3%
Arctic Wolf	Q1 2022 Incident Response Insights from Tetra Defense	82%	External Vulnerabilities 57% RDP 25%
Group-IB	Ransomware Uncovered 2021/2022	68%	External remote services 47% Exploit public-facing applications 21%
IBM	X-Force Threat Intelligence Index 2022	53%	Vulnerability exploitation 47% Stolen credentials 3% Brute force 3%

For more details : http://jsac.jpcert.or.jp/archive/2023/pdf/JSAC2023_1_7_sejiyama_en.pdf

Another major attack vector

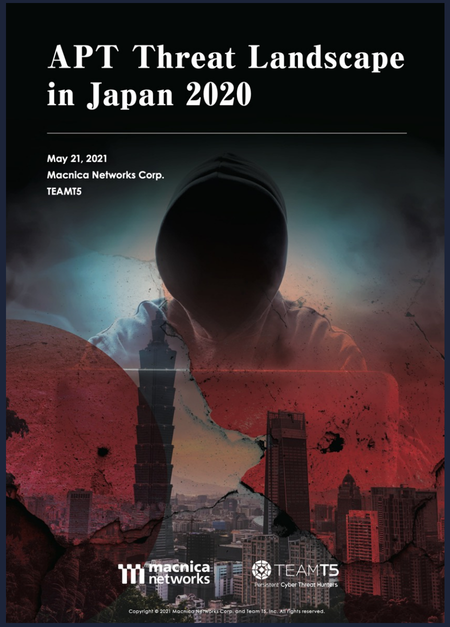
- Mustang Panda
- UNC4191
- UNC4698
- **TA410**

etc..

whoami

- Hiroshi Takeuchi
  - Security Researcher at MACNICA Security Research Center
  - Malware analysis, Incident Response

- Some research publications
  - *Shedding Light on Shadow(PAD) Components* (Mandiant CDS 2021)
  - *Tracking rapid evolutions? Copycat? Of an APT RAT in Asia* (VB2020)
  - *APT Threat Landscape in Japan* (Annual Report)



APT Threat Landscape in Japan 2020

May 21, 2021

Macnica Networks Corp.

TEAMT5

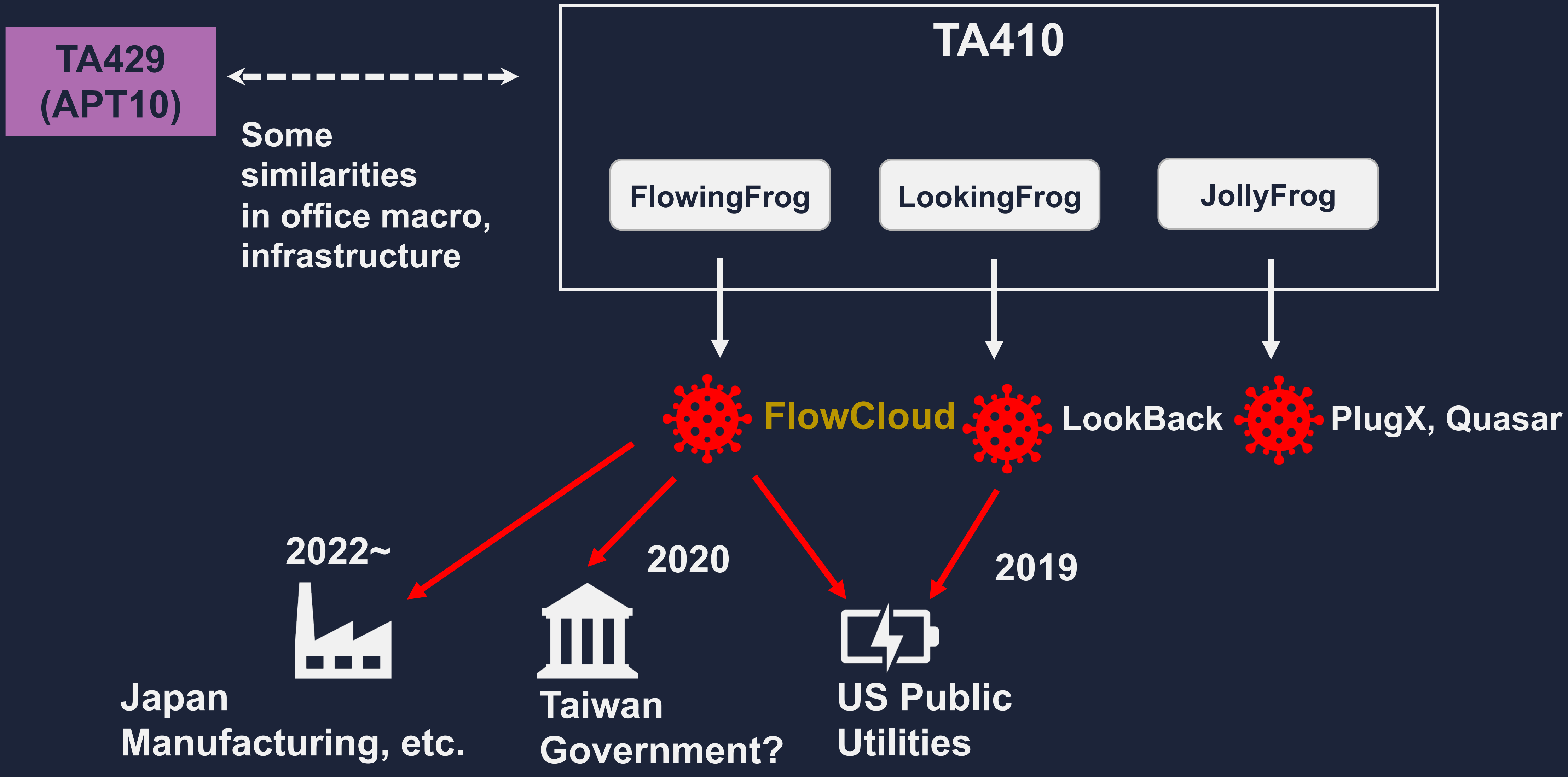
Copyright © 2021 Macnica Networks Corp. and Team T5, Inc. All rights reserved.

Agenda

1. TA410: FlowCloud
2. Operation “USBFlowing”
3. Deep Dive into fcClinetDll Code
4. Developer Profiling
5. Conclusion

TA410: FlowCloud

TA410 activity



TA429 (APT10)

Some similarities in office macro, infrastructure

TA410

FlowingFrog

LookingFrog

JollyFrog

FlowCloud

LookBack

PlugX, Quasar

2022~

Japan Manufacturing, etc.

2020

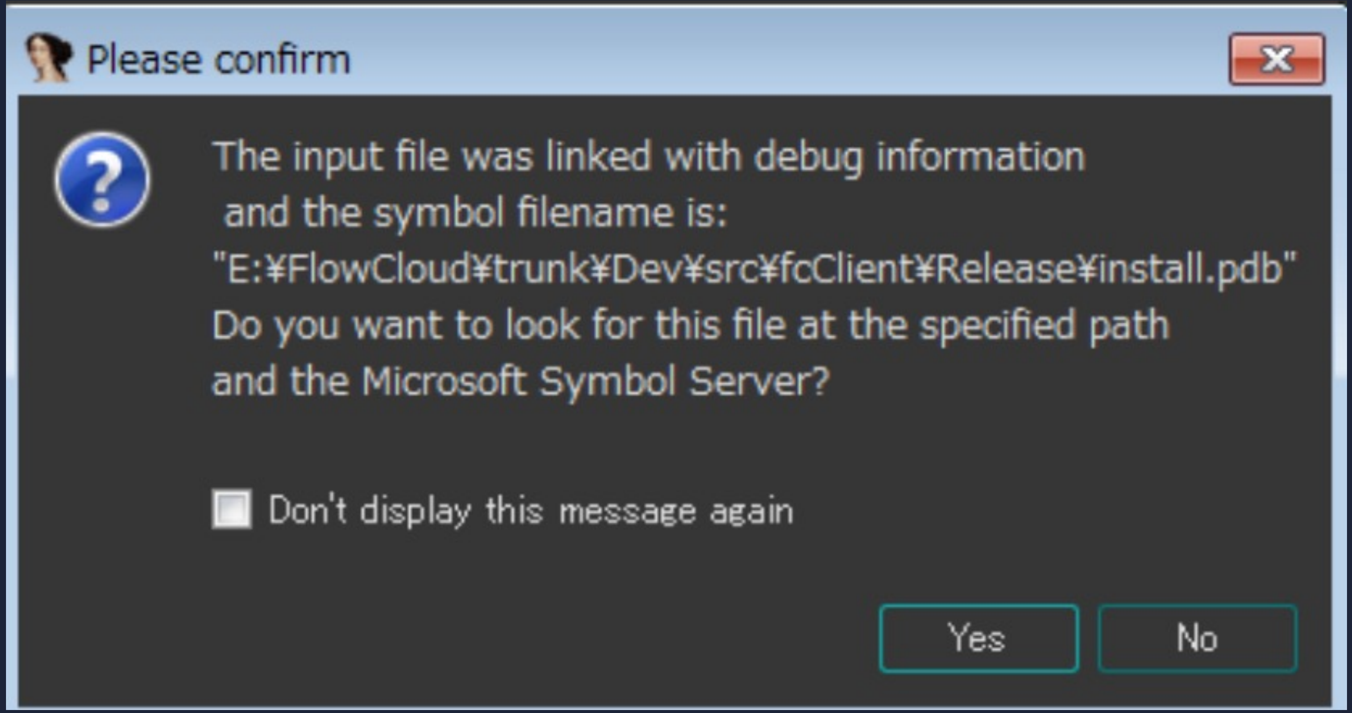
Taiwan Government?

2019

US Public Utilities

FlowCloud

- RAT providing many features, file manipulation, screen capture, recording and rootkit driver.
- Developed by C or C/C++, using open-source libraries, Protocol Buffers, Boost, Zthread
- Installation chain and execution flow is complicated



Please confirm

The input file was linked with debug information
and the symbol filename is:
"E:¥FlowCloud¥trunk¥Dev¥src¥fcClient¥Release¥install.pdb"
Do you want to look for this file at the specified path
and the Microsoft Symbol Server?

□ Don't display this message again

Observed FlowCloud versions

Version	id_prefix	Compile date (UTC) *1	Language & Open Source Library
4.1.3	NN913	Mar 21 02:38:47 2019	C, Protocol Buffers, SQLite
5.0.1	N/A	Sep 26 04:18:48 2016	C++, Protocol Buffers, Boost, ZThread, SQLite
5.0.2	N/A	N/A	Only public information and no sample available *2
5.0.3	N/A	Tue Jul 06 09:12:35 2021	C++, Protocol Buffers, Boost, ZThread, SQLite
5.0.5	N/A	N/A	Only public information and no sample available *3
5.0.7	N/A	Oct 28 05:11:25 2021	C++, Protocol Buffers, Boost, ZThread, SQLite
5.0.8	20220814, 220823	May 25 07:37:08 2022	C++, Protocol Buffers, Boost, ZThread, SQLite
6.0.0	N/A	Feb 15 09:34:54 2023	C++, Protocol Buffers, Boost, ZThread, SQLite

*1 Compile Date of other samples than 4.1.3 is XXXModule_func.dll. We believe it is the most confident from our observation.

*2 https://www.welivesecurity.com/2022/04/27/lookback-ta410-umbrella-cyberespionage-ttps-activity/

*3 https://jp.security.ntt/tech_blog/102ifpu

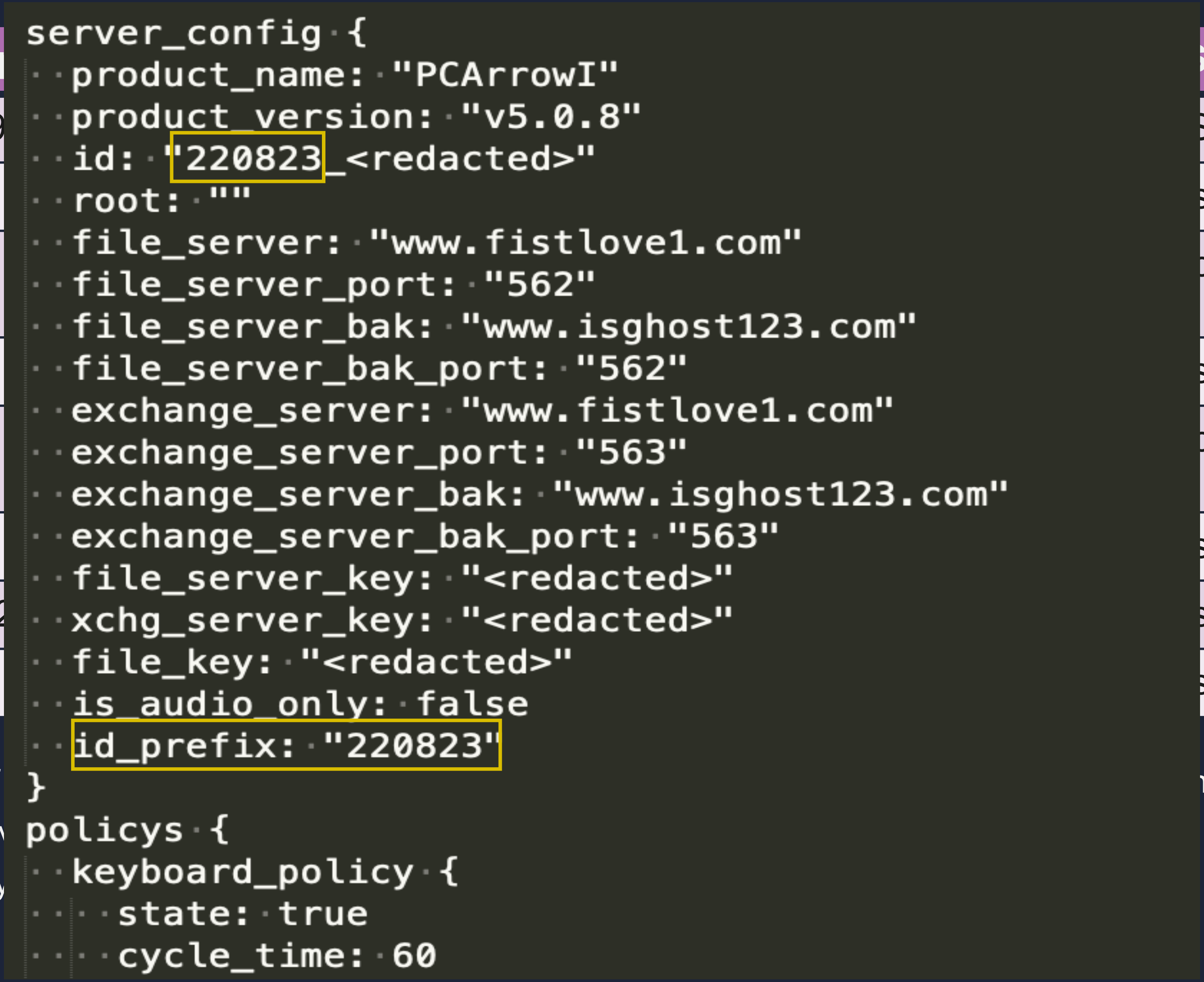
Observed FlowCloud versions

Version	id_	ource Library
4.1.3	NN9	SQLite
5.0.1	N/A	s, Boost, ZThread, SQLite
5.0.2	N/A	on and no sample available
5.0.3	N/A	s, Boost, ZThread, SQLite
5.0.5	N/A	on and no sample available
5.0.7	N/A	s, Boost, ZThread, SQLite
5.0.8	2022	s, Boost, ZThread, SQLite
6.0.0	N/A	s, Boost, ZThread, SQLite

*1 Compile Date of … nt from our observation.

*2 https://www.weliv

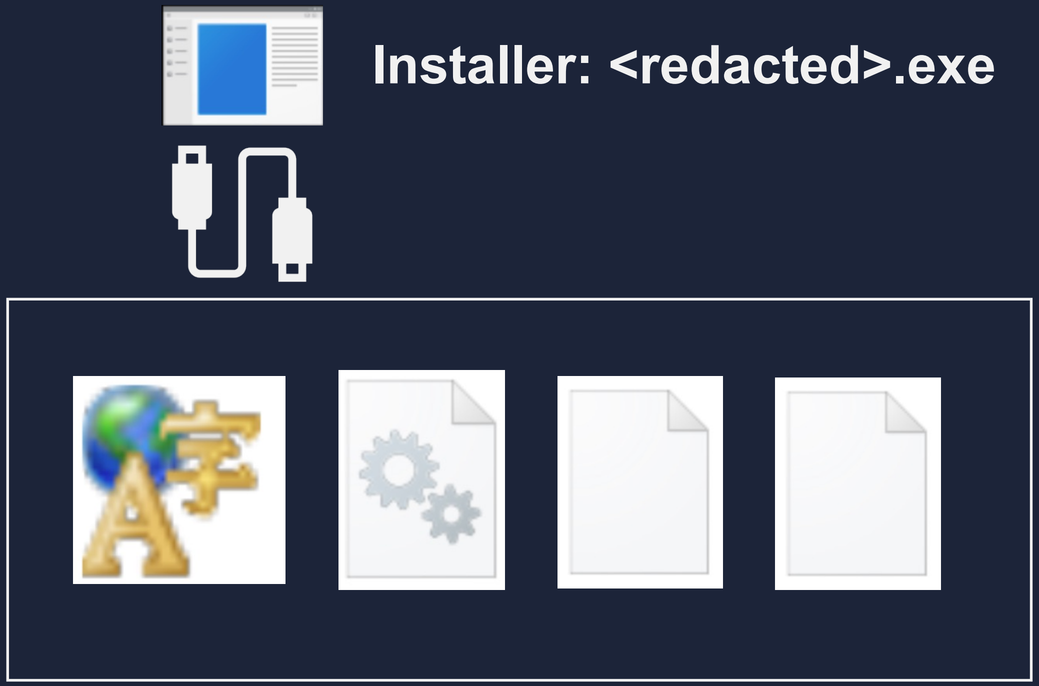
*3 https://jp.security

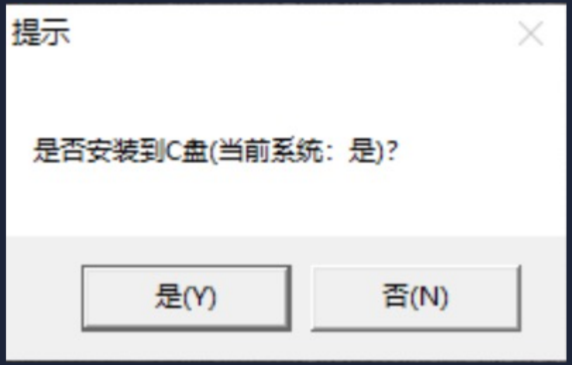


```
server_config {
  product_name: "PCArrowI"
  product_version: "v5.0.8"
  id: "220823_<redacted>"
  root: ""
  file_server: "www.fistlove1.com"
  file_server_port: "562"
  file_server_bak: "www.isghost123.com"
  file_server_bak_port: "562"
  exchange_server: "www.fistlove1.com"
  exchange_server_port: "563"
  exchange_server_bak: "www.isghost123.com"
  exchange_server_bak_port: "563"
  file_server_key: "<redacted>"
  xchg_server_key: "<redacted>"
  file_key: "<redacted>"
  is_audio_only: false
  id_prefix: "220823"
}
policys {
  keyboard_policy {
    state: true
    cycle_time: 60
```

Operation “USBFlowing”

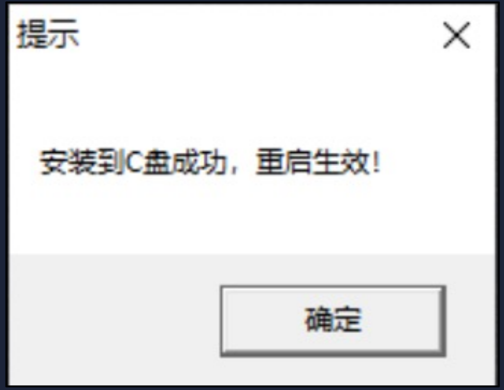
Installation Chain


Installer: <redacted>.exe


提示

是否安装到C盘(当前系统：是)?

Do you install in C Drive? (Default: Yes)

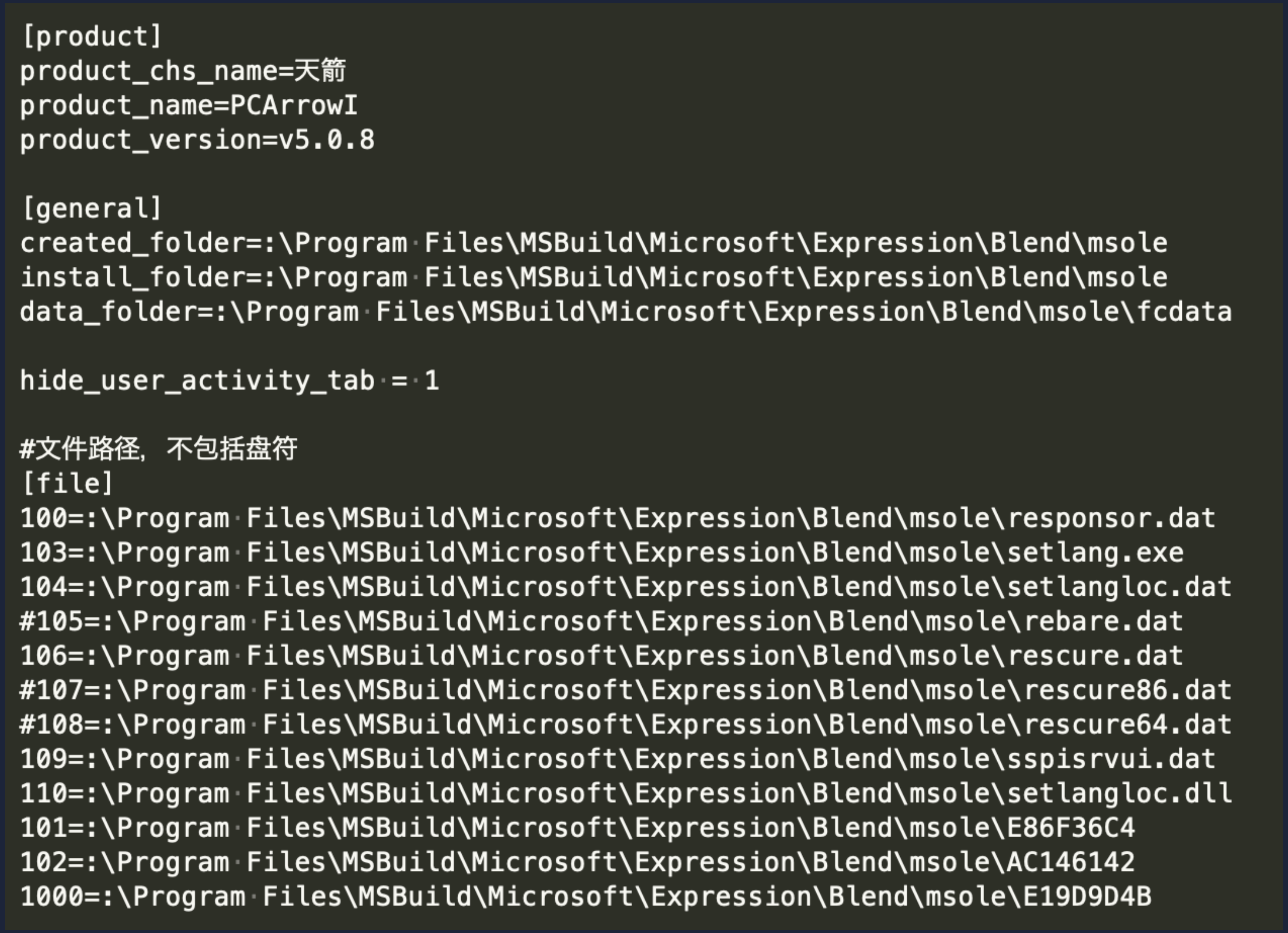

提示

安装到C盘成功，重启生效！

Installation completed, Reboot to take effect!

Installer in USB deploys FlowCloud components in connected device

Install configuration

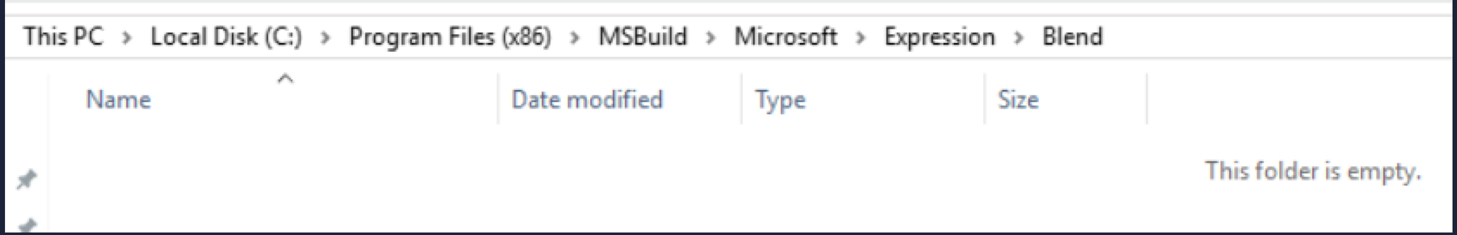


```
[product]
product_chs_name=天箭
product_name=PCArrowI
product_version=v5.0.8

[general]
created_folder=:\Program Files\MSBuild\Microsoft\Expression\Blend\msole
install_folder=:\Program Files\MSBuild\Microsoft\Expression\Blend\msole
data_folder=:\Program Files\MSBuild\Microsoft\Expression\Blend\msole\fcdata

hide_user_activity_tab = 1

#文件路径，不包括盘符
[file]
100=:\Program Files\MSBuild\Microsoft\Expression\Blend\msole\responsor.dat
103=:\Program Files\MSBuild\Microsoft\Expression\Blend\msole\setlang.exe
104=:\Program Files\MSBuild\Microsoft\Expression\Blend\msole\setlangloc.dat
#105=:\Program Files\MSBuild\Microsoft\Expression\Blend\msole\rebare.dat
106=:\Program Files\MSBuild\Microsoft\Expression\Blend\msole\rescure.dat
#107=:\Program Files\MSBuild\Microsoft\Expression\Blend\msole\rescure86.dat
#108=:\Program Files\MSBuild\Microsoft\Expression\Blend\msole\rescure64.dat
109=:\Program Files\MSBuild\Microsoft\Expression\Blend\msole\sspisrvui.dat
110=:\Program Files\MSBuild\Microsoft\Expression\Blend\msole\setlangloc.dll
101=:\Program Files\MSBuild\Microsoft\Expression\Blend\msole\E86F36C4
102=:\Program Files\MSBuild\Microsoft\Expression\Blend\msole\AC146142
1000=:\Program Files\MSBuild\Microsoft\Expression\Blend\msole\E19D9D4B
```

Anti-forensic technique

- The directory (msole) containing FlowCloud components is hidden with system and hidden attribution. We need to remove them to collect artifacts.

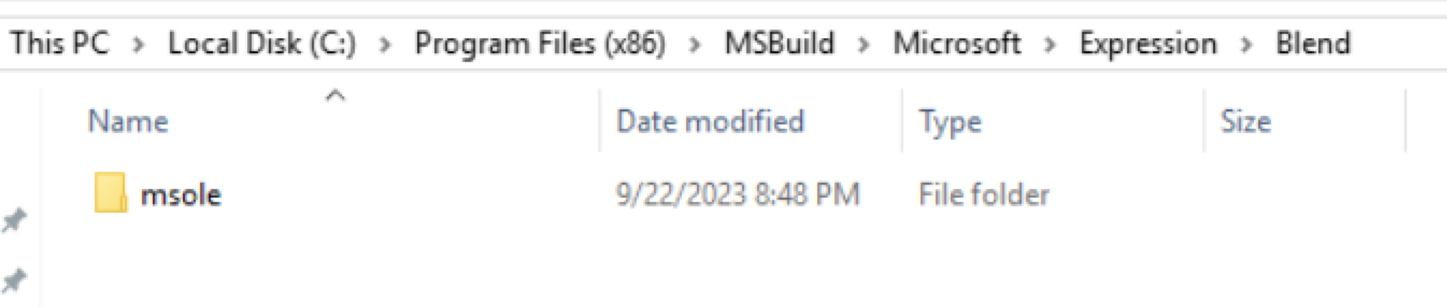
```
cd C:¥Program Files (x86)¥MSBuild¥Microsoft¥Expression¥Blend
attrib msole -s -h
```



This PC › Local Disk (C:) › Program Files (x86) › MSBuild › Microsoft › Expression › Blend

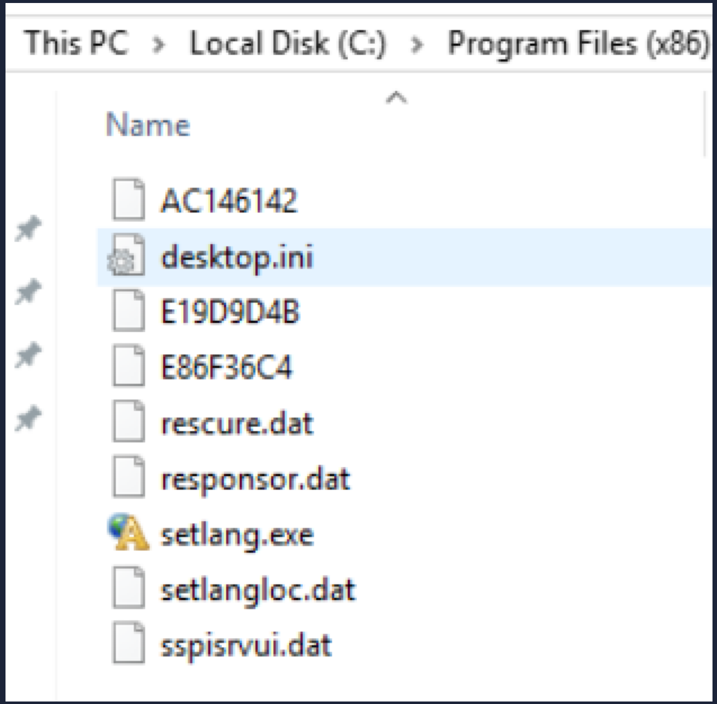
Name	Date modified	Type	Size

This folder is empty.



This PC › Local Disk (C:) › Program Files (x86) › MSBuild › Microsoft › Expression › Blend

Name	Date modified	Type	Size
msole	9/22/2023 8:48 PM	File folder	



This PC › Local Disk (C:) › Program Files (x86)

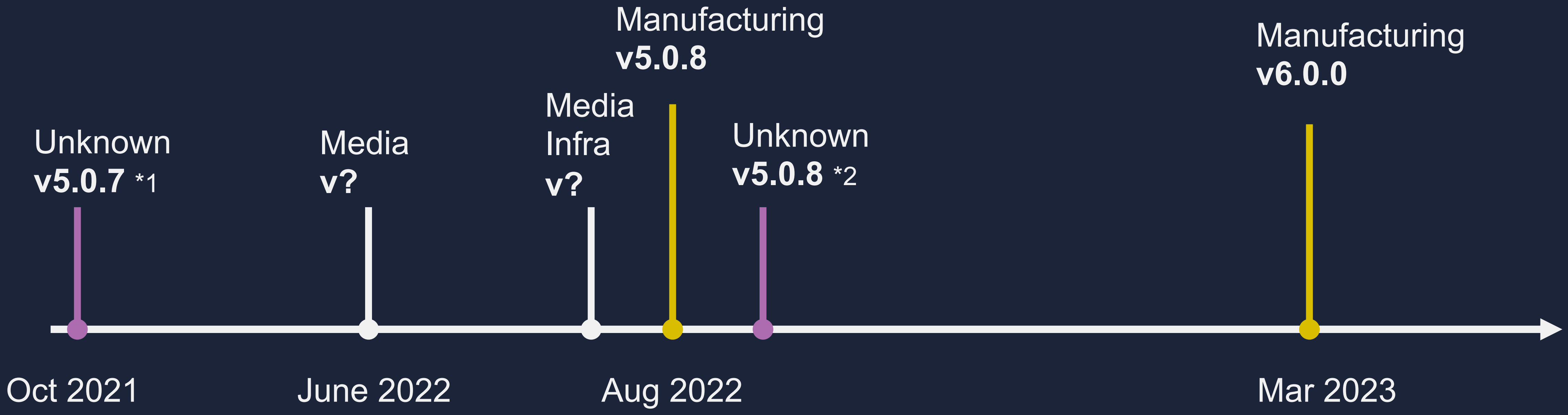
Name

- AC146142
- desktop.ini
- E19D9D4B
- E86F36C4
- rescure.dat
- responsor.dat
- setlang.exe
- setlangloc.dat
- sspisrvui.dat

Why USB?

1. Targeting air-gapped network
2. Easy to handle for infection
   - Auto propagation (Worm)
   - Legitimate software camouflage (Social Engineering)
   - Manual operation (Classic espionage)

Operation USBFlowing Timeline



● Our sensors

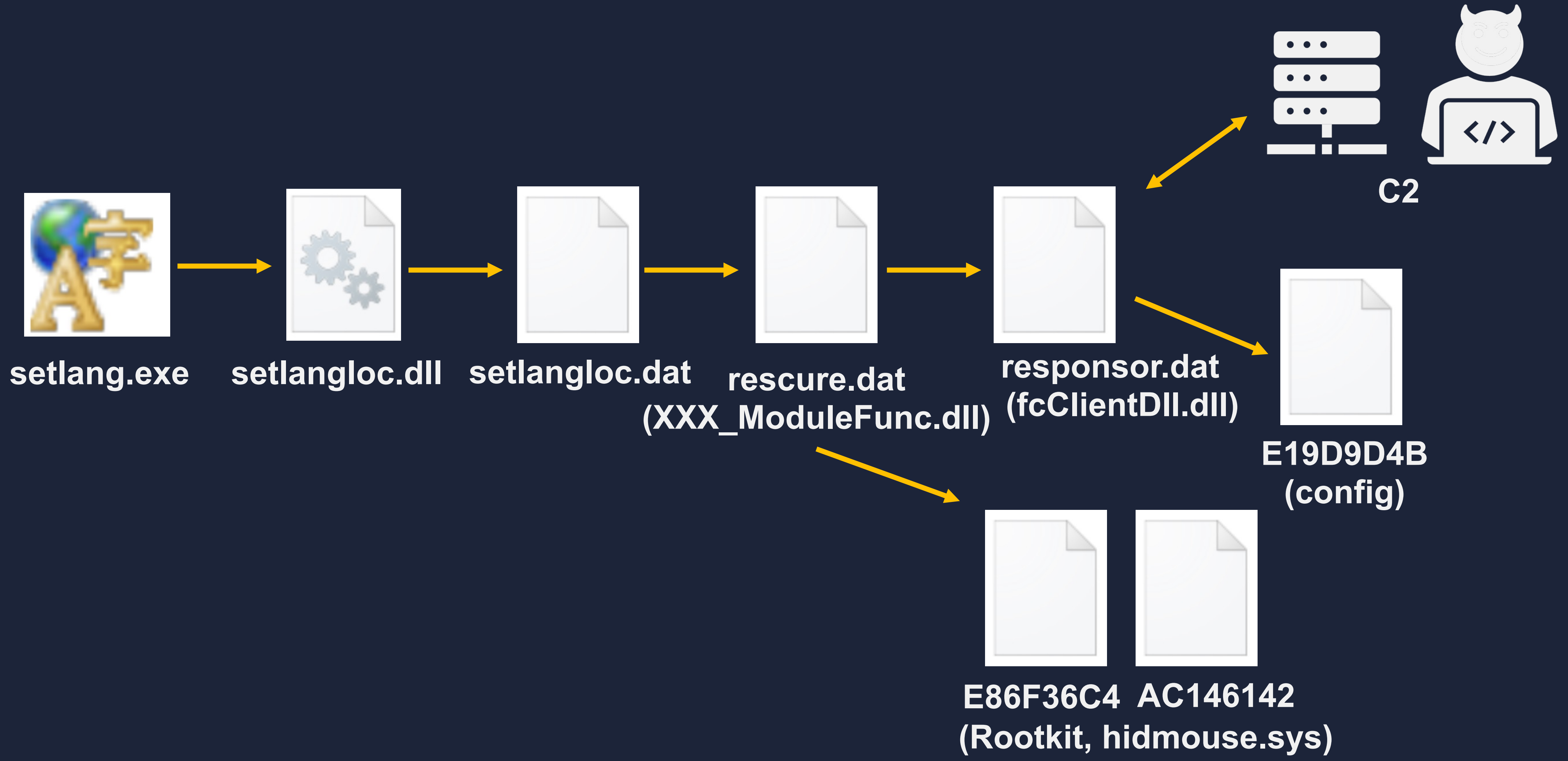
○ Other vendors (https://jp.security.ntt/tech_blog/102ifpu)

● Malware Repository

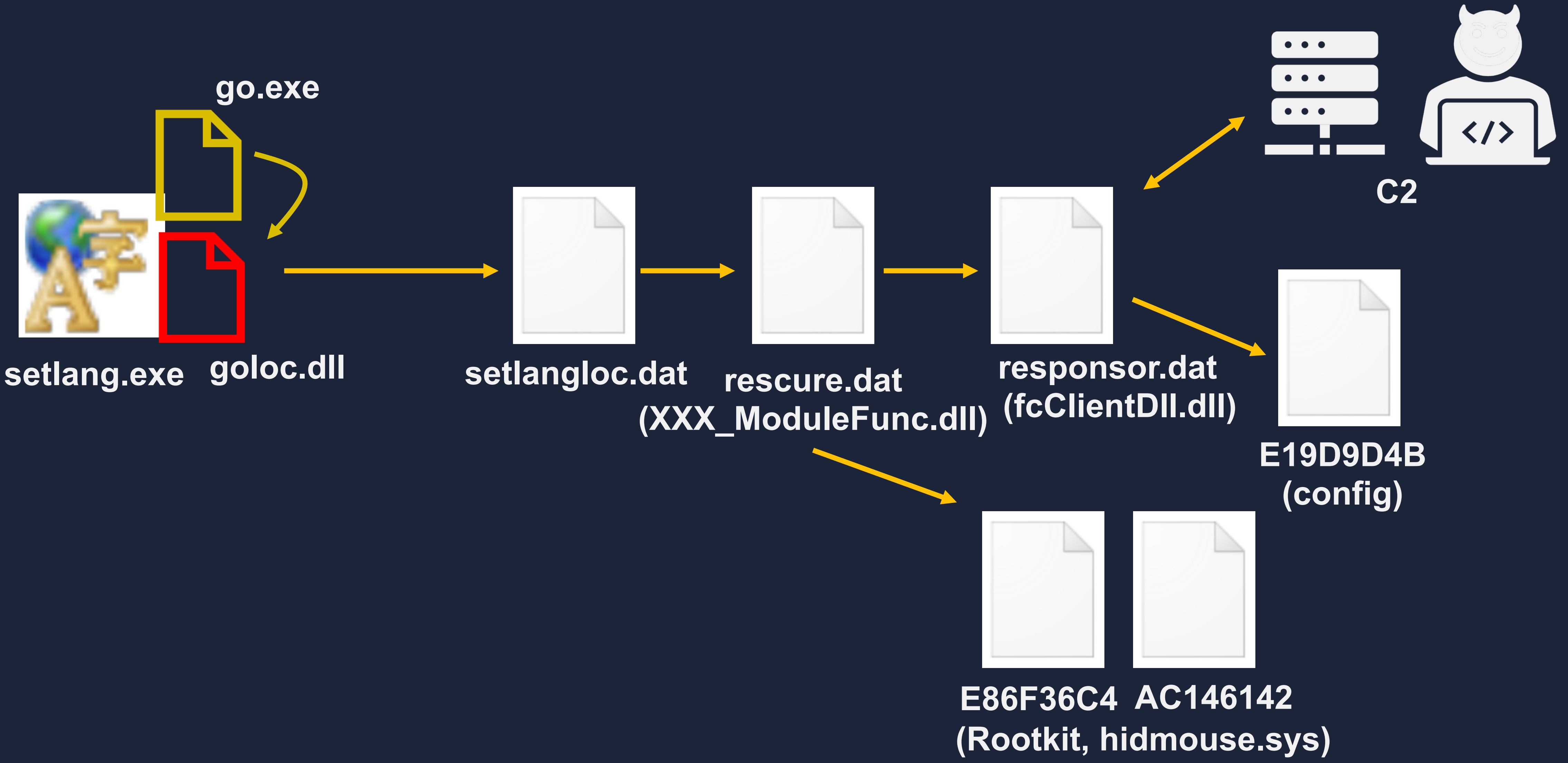
*1 presumption from compile time-stamp of XXXModule_func.dll (loader)
*2 presumption from id prefix of configuration

Japanese organizations' branch offices in China were targeted.

FlowCloud v5.0.7, v5.0.8 execution flow



FlowCloud v6.0.0 execution flow



go.exe

setlang.exe goloc.dll

setlangloc.dat

rescure.dat
(XXX_ModuleFunc.dll)

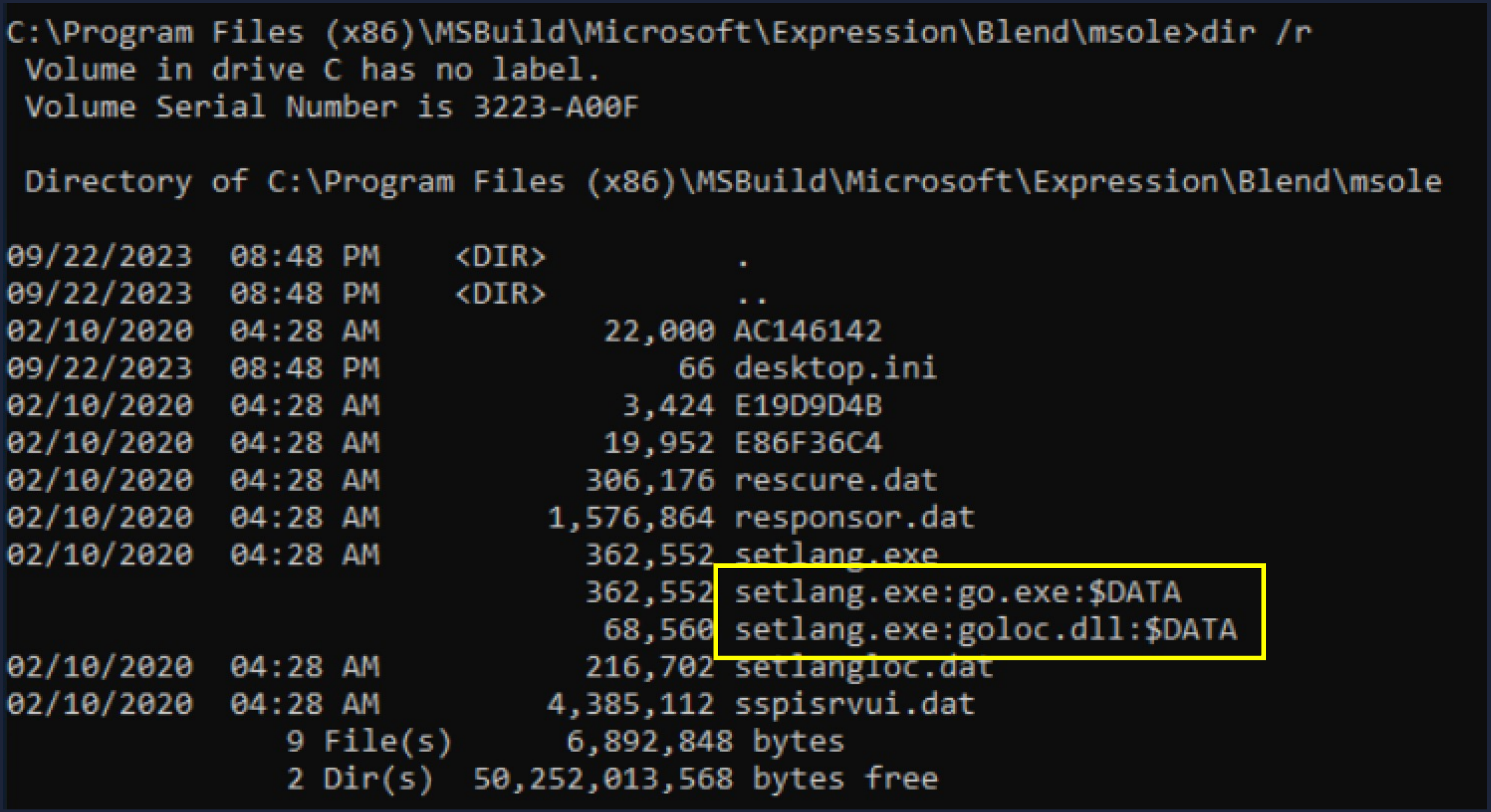
responsor.dat
(fcClientDll.dll)

C2

E19D9D4B
(config)

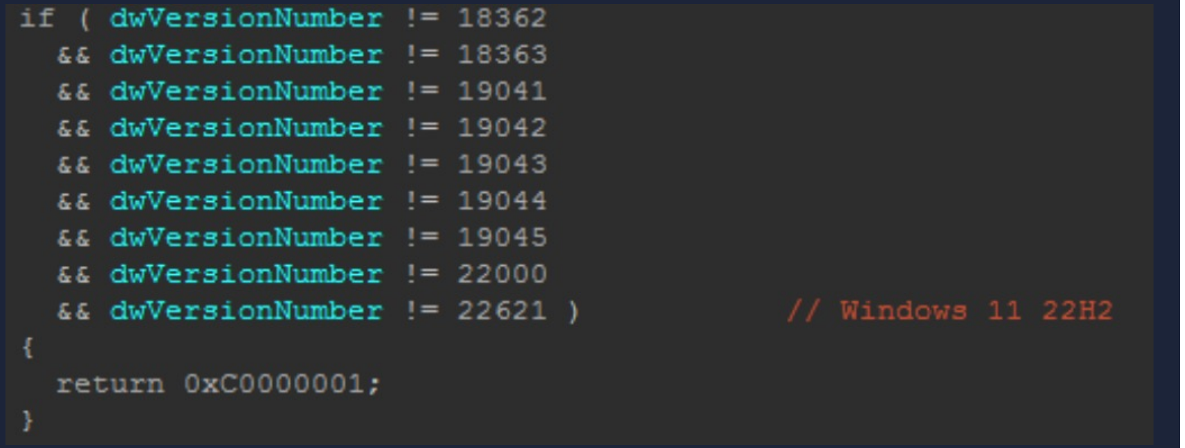
E86F36C4 AC146142
(Rootkit, hidmouse.sys)

Hide Artifacts: NTFS File Attributes

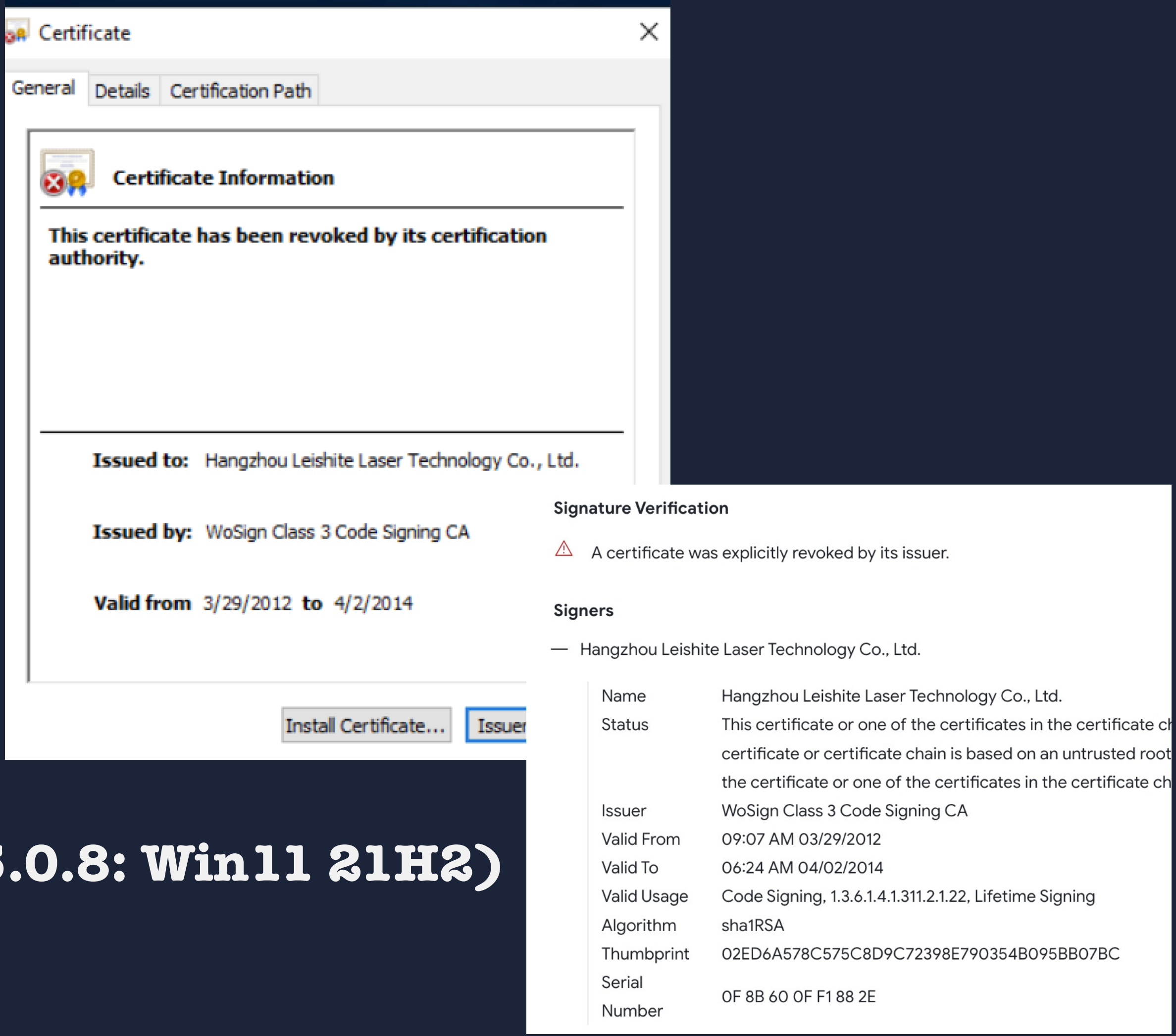


```
C:\Program Files (x86)\MSBuild\Microsoft\Expression\Blend\msole>dir /r
 Volume in drive C has no label.
 Volume Serial Number is 3223-A00F

 Directory of C:\Program Files (x86)\MSBuild\Microsoft\Expression\Blend\msole

09/22/2023  08:48 PM    <DIR>          .
09/22/2023  08:48 PM    <DIR>          ..
02/10/2020  04:28 AM            22,000 AC146142
09/22/2023  08:48 PM                66 desktop.ini
02/10/2020  04:28 AM             3,424 E19D9D4B
02/10/2020  04:28 AM            19,952 E86F36C4
02/10/2020  04:28 AM           306,176 rescure.dat
02/10/2020  04:28 AM         1,576,864 responsor.dat
02/10/2020  04:28 AM           362,552 setlang.exe
                               362,552 setlang.exe:go.exe:$DATA
                                68,560 setlang.exe:goloc.dll:$DATA
02/10/2020  04:28 AM           216,702 setlangloc.dat
02/10/2020  04:28 AM         4,385,112 sspisrvui.dat
               9 File(s)      6,892,848 bytes
               2 Dir(s)  50,252,013,568 bytes free
```

Rootkit Driver: hidmouse.sys



```
if ( dwVersionNumber != 18362
  && dwVersionNumber != 18363
  && dwVersionNumber != 19041
  && dwVersionNumber != 19042
  && dwVersionNumber != 19043
  && dwVersionNumber != 19044
  && dwVersionNumber != 19045
  && dwVersionNumber != 22000
  && dwVersionNumber != 22621 )          // Windows 11 22H2
{
  return 0xC0000001;
}
```



Certificate Information

This certificate has been revoked by its certification authority.

Issued to: Hangzhou Leishite Laser Technology Co., Ltd.

Issued by: WoSign Class 3 Code Signing CA

Valid from 3/29/2012 to 4/2/2014

Signature Verification

⚠ A certificate was explicitly revoked by its issuer.

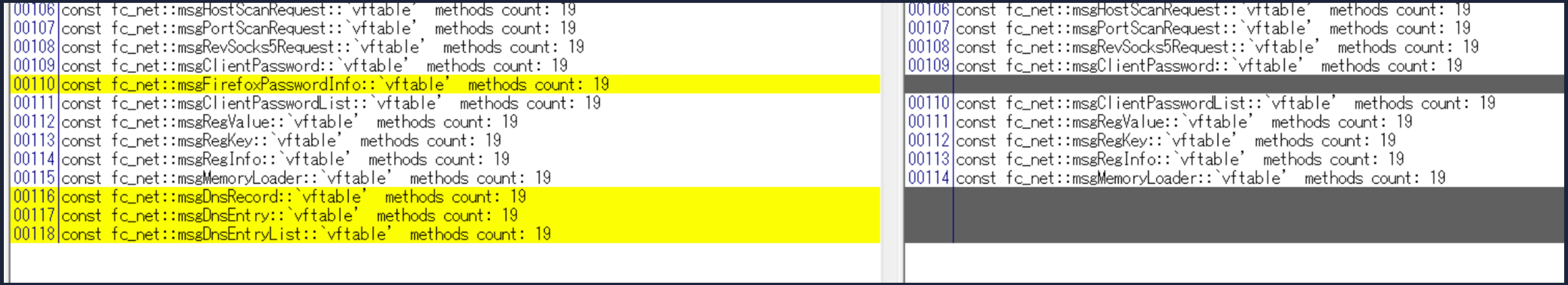
Signers

— Hangzhou Leishite Laser Technology Co., Ltd.

Name	Hangzhou Leishite Laser Technology Co., Ltd.
Status	This certificate or one of the certificates in the certificate ch certificate or certificate chain is based on an untrusted root the certificate or one of the certificates in the certificate ch
Issuer	WoSign Class 3 Code Signing CA
Valid From	09:07 AM 03/29/2012
Valid To	06:24 AM 04/02/2014
Valid Usage	Code Signing, 1.3.6.1.4.1.311.2.1.22, Lifetime Signing
Algorithm	sha1RSA
Thumbprint	02ED6A578C575C8D9C72398E790354B095BB07BC
Serial Number	0F 8B 60 0F F1 88 2E

- * v6.0.0 supports Windows 11 22H2 (v5.0.8: Win11 21H2)
- * Same stolen certificate has been used

fcClinetDll: v6.0.0 vs v5.0.8 Diff



```
00106 const fc_net::msgHostScanRequest::`vftable’  methods count: 19
00107 const fc_net::msgPortScanRequest::`vftable’  methods count: 19
00108 const fc_net::msgRevSocks5Request::`vftable’  methods count: 19
00109 const fc_net::msgClientPassword::`vftable’  methods count: 19
00110 const fc_net::msgFirefoxPasswordInfo::`vftable’  methods count: 19
00111 const fc_net::msgClientPasswordList::`vftable’  methods count: 19
00112 const fc_net::msgRegValue::`vftable’  methods count: 19
00113 const fc_net::msgRegKey::`vftable’  methods count: 19
00114 const fc_net::msgRegInfo::`vftable’  methods count: 19
00115 const fc_net::msgMemoryLoader::`vftable’  methods count: 19
00116 const fc_net::msgDnsRecord::`vftable’  methods count: 19
00117 const fc_net::msgDnsEntry::`vftable’  methods count: 19
00118 const fc_net::msgDnsEntryList::`vftable’  methods count: 19
```

```
00106 const fc_net::msgHostScanRequest::`vftable’  methods count: 19
00107 const fc_net::msgPortScanRequest::`vftable’  methods count: 19
00108 const fc_net::msgRevSocks5Request::`vftable’  methods count: 19
00109 const fc_net::msgClientPassword::`vftable’  methods count: 19

00110 const fc_net::msgClientPasswordList::`vftable’  methods count: 19
00111 const fc_net::msgRegValue::`vftable’  methods count: 19
00112 const fc_net::msgRegKey::`vftable’  methods count: 19
00113 const fc_net::msgRegInfo::`vftable’  methods count: 19
00114 const fc_net::msgMemoryLoader::`vftable’  methods count: 19
```

v6.0.0 **v5.0.8**

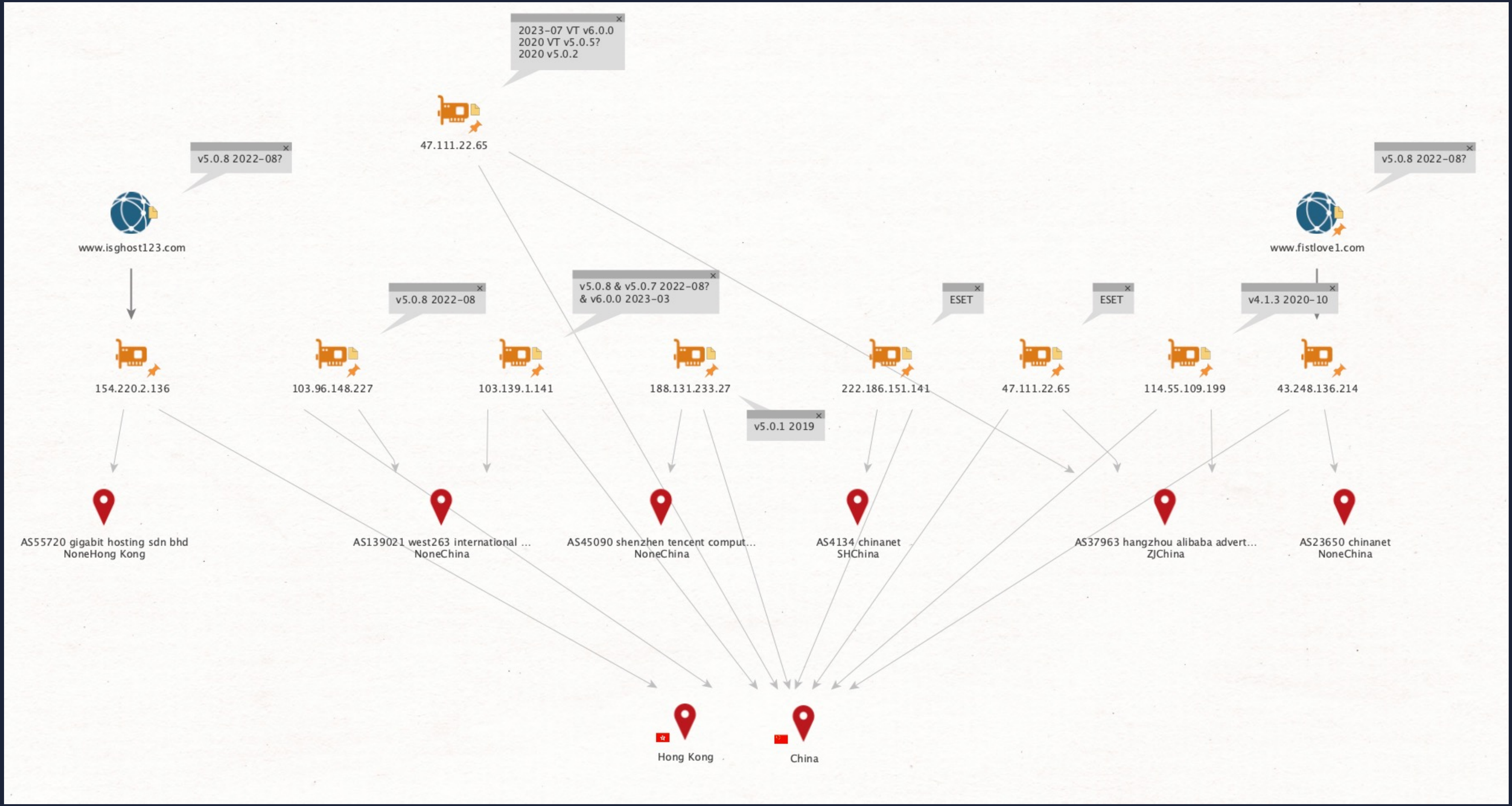
Some new plugin modules are supported in 6.0.0.

FlowCloud v6.0.0 variant

- Uploaded to VirusTotal in July 2023
  - 6db73d48041a069d42dc8625c59754cba2760189b9a6412a3986411cd3a0e573
- rescure.dat (XXXModule_func.dll) is VMProtected (Not found in the field)
- New plugin classes implemented in v6.0.0 are missing in this file (msgFireFoxPasswordInfo, etc)
- Compile date of XXXModule_func.dll is Dec 20 06:53:36 2022

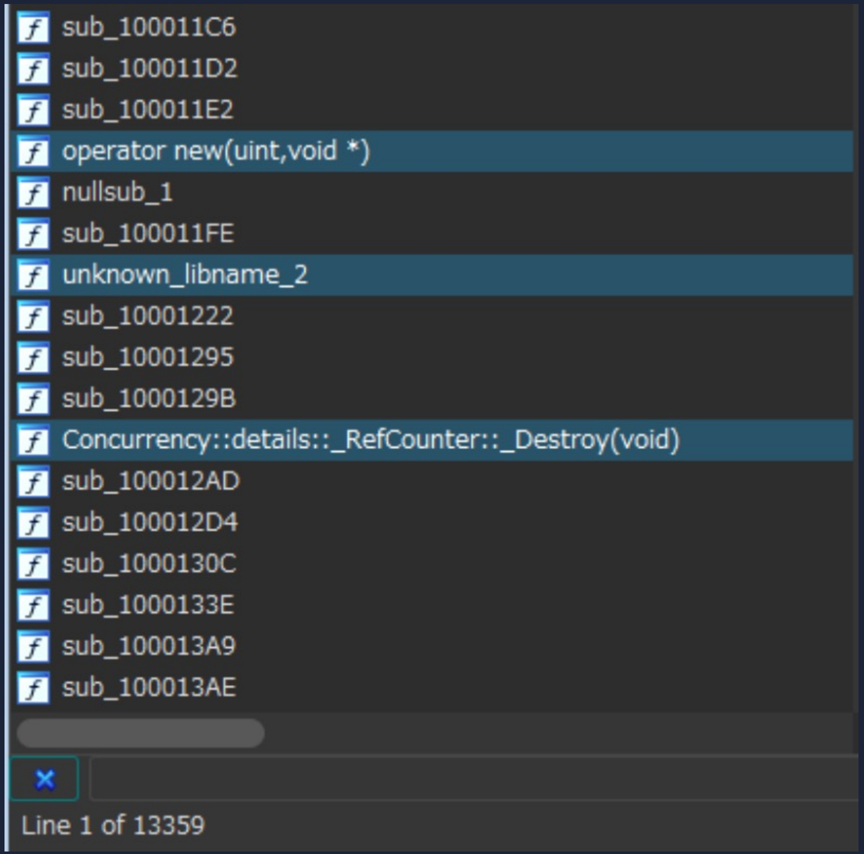
This sample is probably testing purpose built.

C2 Infrastructure

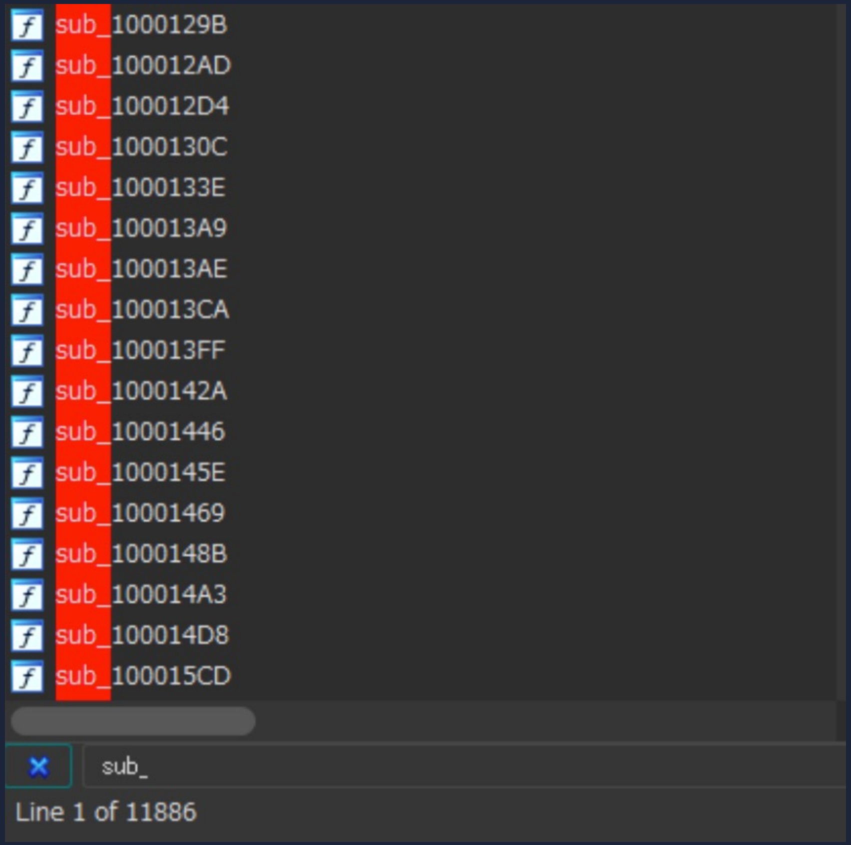


Deep Dive into fcClinetDll Code

Starting point of journey



```
sub_100011C6
sub_100011D2
sub_100011E2
operator new(uint,void *)
nullsub_1
sub_100011FE
unknown_libname_2
sub_10001222
sub_10001295
sub_1000129B
Concurrency::details::_RefCounter::_Destroy(void)
sub_100012AD
sub_100012D4
sub_1000130C
sub_1000133E
sub_100013A9
sub_100013AE

Line 1 of 13359
```

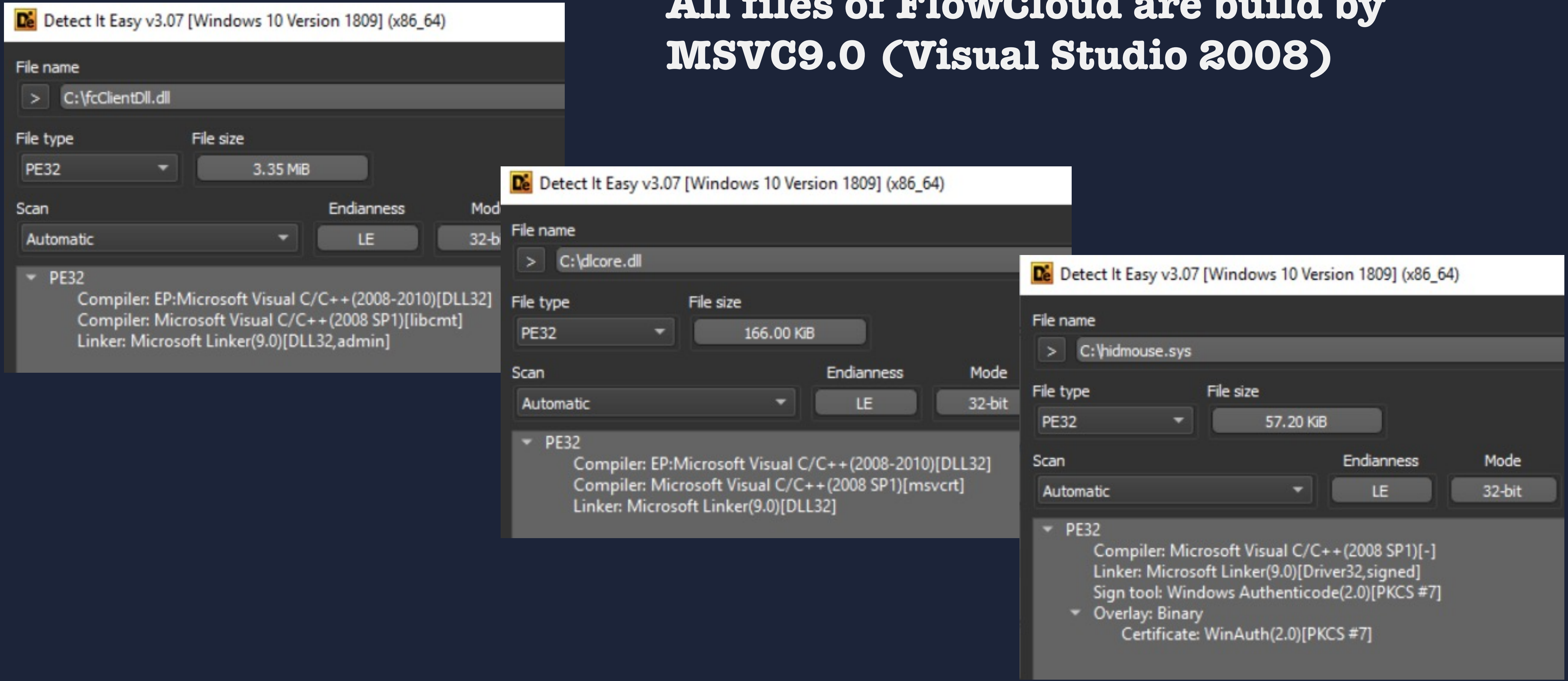


```
sub_1000129B
sub_100012AD
sub_100012D4
sub_1000130C
sub_1000133E
sub_100013A9
sub_100013AE
sub_100013CA
sub_100013FF
sub_1000142A
sub_10001446
sub_1000145E
sub_10001469
sub_1000148B
sub_100014A3
sub_100014D8
sub_100015CD

sub_
Line 1 of 11886
```

Open-source libraries are linked statically and the number of unidentified functions is 11886. This is big challenge for analysts.

A thing in common

All files of FlowCloud are build by MSVC9.0 (Visual Studio 2008)



Detect It Easy v3.07 [Windows 10 Version 1809] (x86_64)

File name: C:\fcClientDll.dll

File type: PE32

File size: 3.35 MiB

Scan: Automatic

Endianness: LE

Mod 32-b

```
PE32
    Compiler: EP:Microsoft Visual C/C++(2008-2010)[DLL32]
    Compiler: Microsoft Visual C/C++(2008 SP1)[libcmt]
    Linker: Microsoft Linker(9.0)[DLL32,admin]
```

Detect It Easy v3.07 [Windows 10 Version 1809] (x86_64)

File name: C:\dlcore.dll

File type: PE32

File size: 166.00 KiB

Scan: Automatic

Endianness: LE

Mode: 32-bit

```
PE32
    Compiler: EP:Microsoft Visual C/C++(2008-2010)[DLL32]
    Compiler: Microsoft Visual C/C++(2008 SP1)[msvcrt]
    Linker: Microsoft Linker(9.0)[DLL32]
```

Detect It Easy v3.07 [Windows 10 Version 1809] (x86_64)

File name: C:\hidmouse.sys

File type: PE32

File size: 57.20 KiB

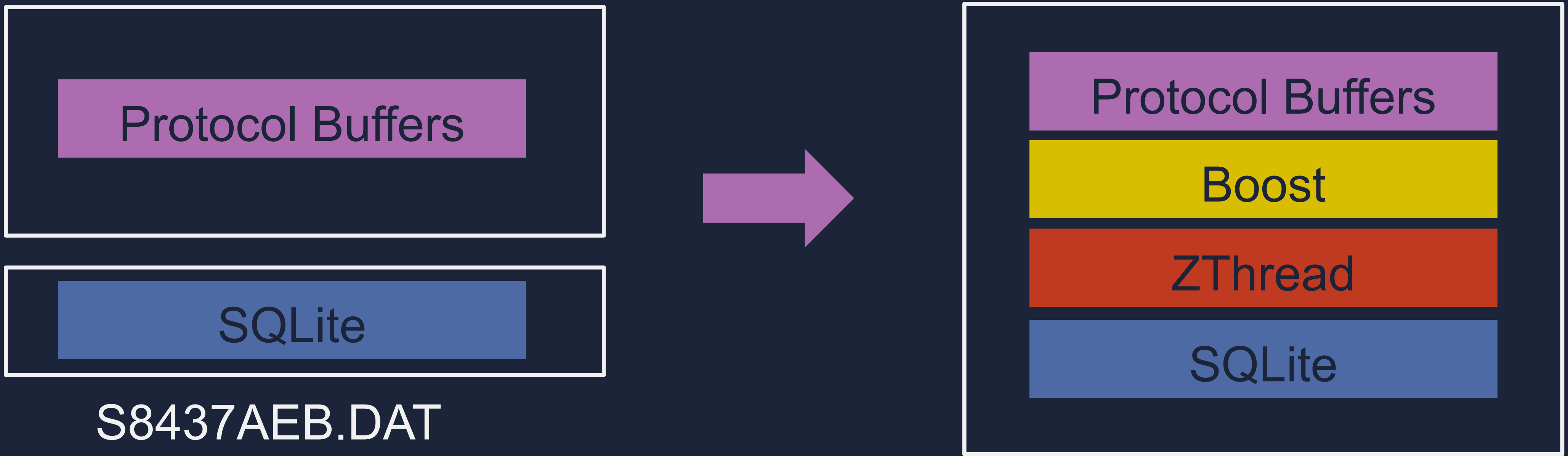
Scan: Automatic

Endianness: LE

Mode: 32-bit

```
PE32
    Compiler: Microsoft Visual C/C++(2008 SP1)[-]
    Linker: Microsoft Linker(9.0)[Driver32,signed]
    Sign tool: Windows Authenticode(2.0)[PKCS #7]
    Overlay: Binary
        Certificate: WinAuth(2.0)[PKCS #7]
```

Open-source library components



Protocol Buffers

SQLite

S8437AEB.DAT

Protocol Buffers

Boost

ZThread

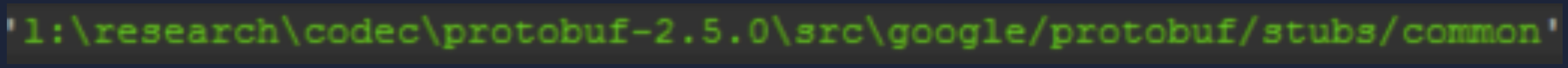
SQLite

4.1.3 (C Language)

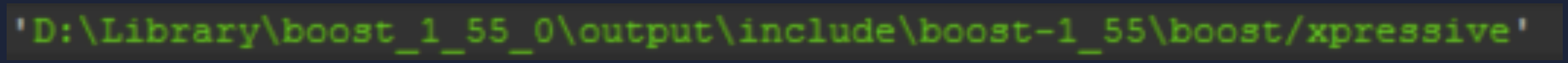
5.0.1~ (C++ Language)

Summary of FlowCloud open-source Library Components

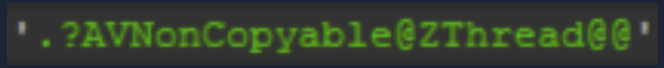
- Protocol Buffers 2.5.0
- boost 1.55.0
- ZThread 2.3.x (Probably 2.3.2)
- SQLite 3.7.16



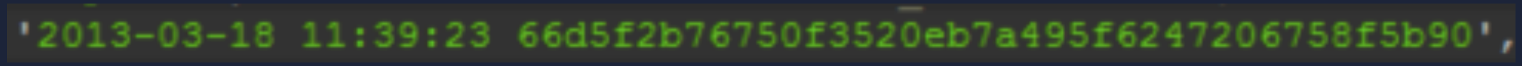
```
'l:\research\codec\protobuf-2.5.0\src\google/protobuf/stubs/common'
```



```
'D:\Library\boost_1_55_0\output\include\boost-1_55\boost/xpressive'
```



```
'.?AVNonCopyable@ZThread@@'
```



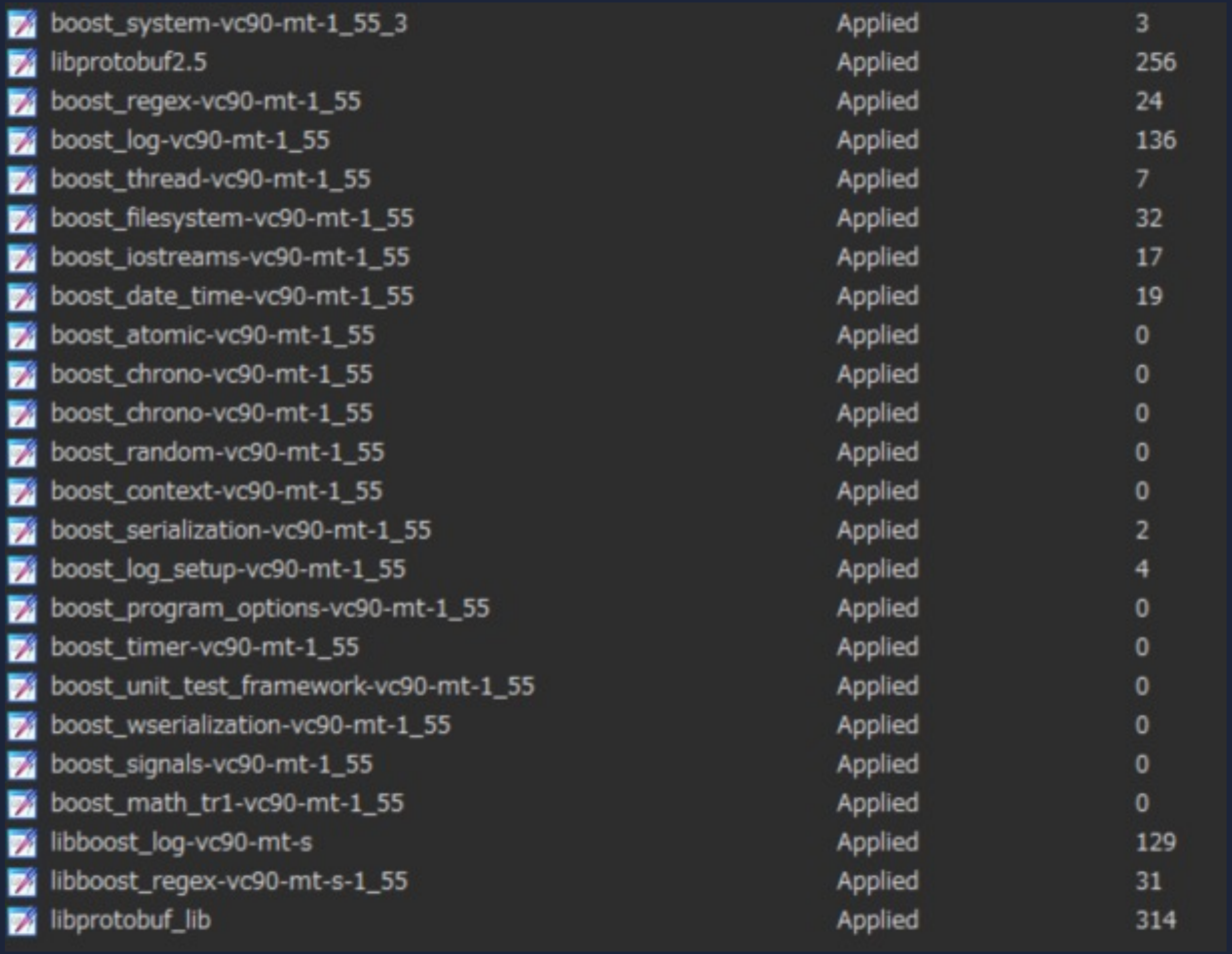
```
'2013-03-18 11:39:23 66d5f2b76750f3520eb7a495f6247206758f5b90',
```

Identify open-source functions: IDA Pro FLIRT

- Build open-source libraries by Microsoft Visual Studio 2008 SP1
  - Protocol Buffers 2.5.0
  - ZThread 2.3.2
  - boost 1.55.0
  - SQLite 3.7.16
- Make FLIRT signature from them
  - IDA 8.0 released make pat file plugin

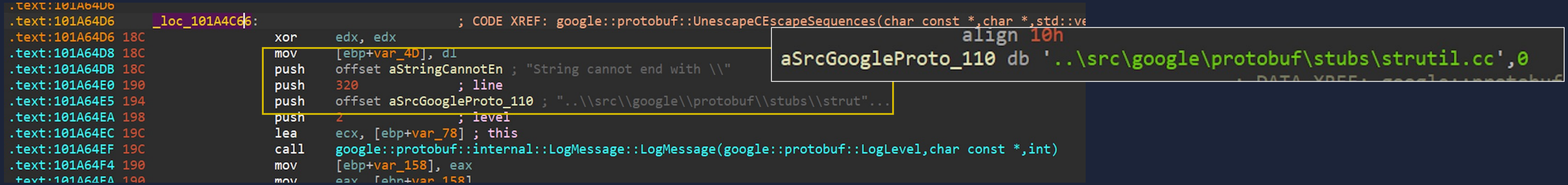
FLIRT sig matching result

- Not good result. Because of STL templates and Compiler Optimization.



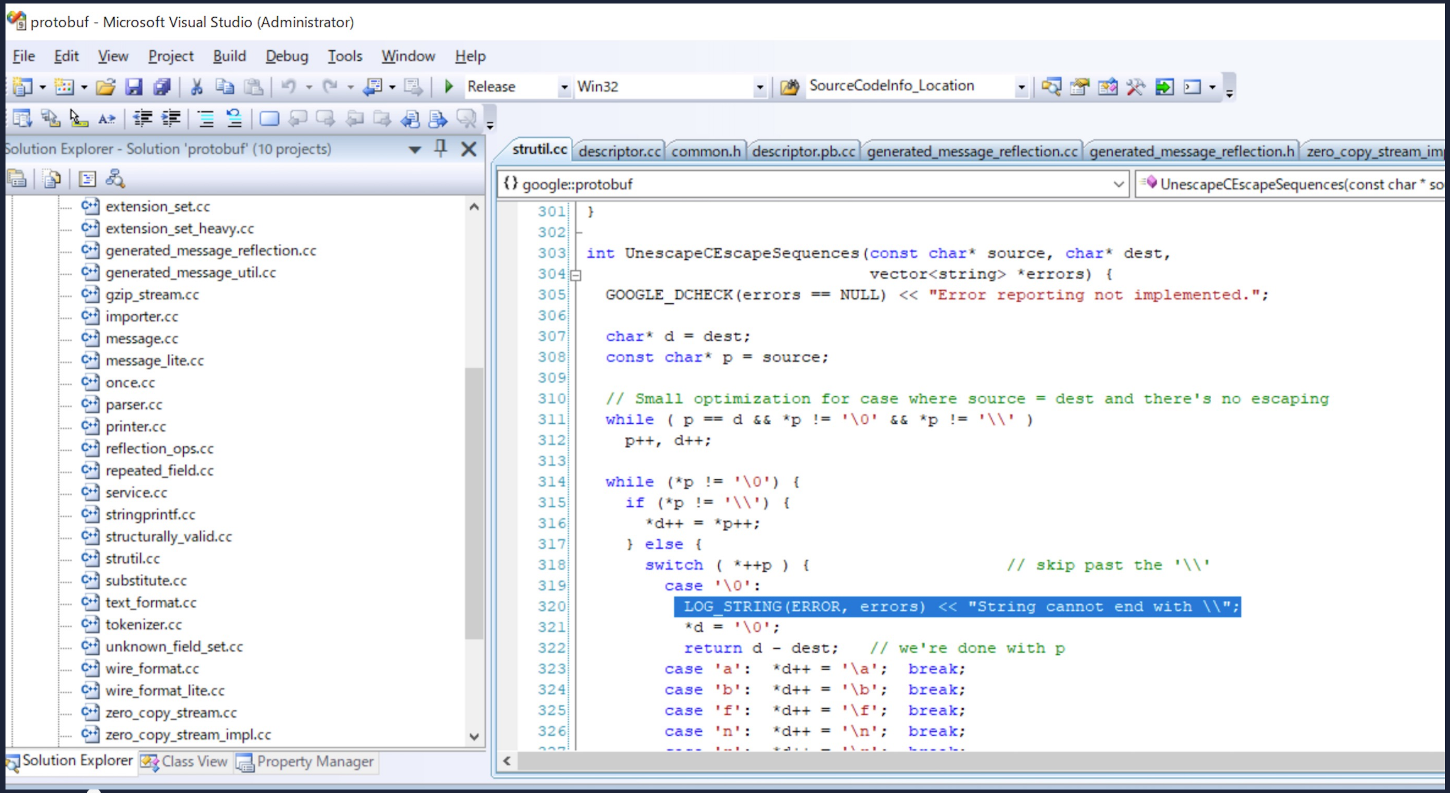
boost_system-vc90-mt-1_55_3	Applied	3
libprotobuf2.5	Applied	256
boost_regex-vc90-mt-1_55	Applied	24
boost_log-vc90-mt-1_55	Applied	136
boost_thread-vc90-mt-1_55	Applied	7
boost_filesystem-vc90-mt-1_55	Applied	32
boost_iostreams-vc90-mt-1_55	Applied	17
boost_date_time-vc90-mt-1_55	Applied	19
boost_atomic-vc90-mt-1_55	Applied	0
boost_chrono-vc90-mt-1_55	Applied	0
boost_chrono-vc90-mt-1_55	Applied	0
boost_random-vc90-mt-1_55	Applied	0
boost_context-vc90-mt-1_55	Applied	0
boost_serialization-vc90-mt-1_55	Applied	2
boost_log_setup-vc90-mt-1_55	Applied	4
boost_program_options-vc90-mt-1_55	Applied	0
boost_timer-vc90-mt-1_55	Applied	0
boost_unit_test_framework-vc90-mt-1_55	Applied	0
boost_wserialization-vc90-mt-1_55	Applied	0
boost_signals-vc90-mt-1_55	Applied	0
boost_math_tr1-vc90-mt-1_55	Applied	0
libboost_log-vc90-mt-s	Applied	129
libboost_regex-vc90-mt-s-1_55	Applied	31
libprotobuf_lib	Applied	314

Identify open-source functions: Source code review



```
.text:101A64D6
.text:101A64D6      _loc_101A4C66:                          ; CODE XREF: google::protobuf::UnescapeCEscapeSequences(char const *,char *,std::ve
.text:101A64D6 18C                 xor     edx, edx
.text:101A64D8 18C                 mov     [ebp+var_4D], dl
.text:101A64DB 18C                 push    offset aStringCannotEn ; "String cannot end with \\"
.text:101A64E0 190                 push    320             ; line
.text:101A64E5 194                 push    offset aSrcGoogleProto_110 ; "..\\src\\google\\protobuf\\stubs\\strut"...
.text:101A64EA 198                 push    2               ; level
.text:101A64EC 19C                 lea     ecx, [ebp+var_78] ; this
.text:101A64EF 19C                 call    google::protobuf::internal::LogMessage::LogMessage(google::protobuf::LogLevel,char const *,int)
.text:101A64F4 190                 mov     [ebp+var_158], eax
.text:101A64FA 190                 mov     eax, [ebp+var_158]
```

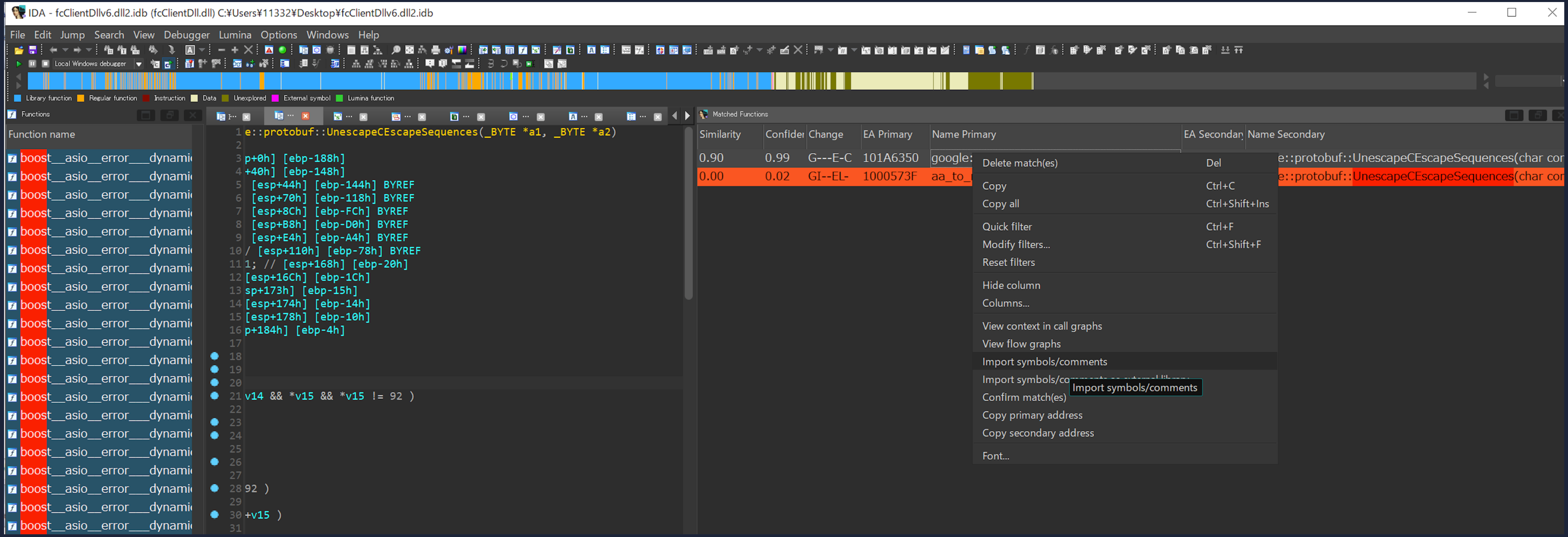
```
                    align 10h
aSrcGoogleProto_110 db '..\src\google\protobuf\stubs\strutil.cc',0
                                        ; DATA XREF: google::protobuf
```



```
}

int UnescapeCEscapeSequences(const char* source, char* dest,
                             vector<string> *errors) {
  GOOGLE_DCHECK(errors == NULL) << "Error reporting not implemented.";

  char* d = dest;
  const char* p = source;

  // Small optimization for case where source = dest and there's no escaping
  while ( p == d && *p != '\0' && *p != '\\' )
    p++, d++;

  while (*p != '\0') {
    if (*p != '\\') {
      *d++ = *p++;
    } else {
      switch ( *++p ) {                    // skip past the '\\'
        case '\0':
          LOG_STRING(ERROR, errors) << "String cannot end with \\";
          *d = '\0';
          return d - dest;   // we're done with p
        case 'a':  *d++ = '\a';  break;
        case 'b':  *d++ = '\b';  break;
        case 'f':  *d++ = '\f';  break;
        case 'n':  *d++ = '\n';  break;
```

We can identify some functions from left debug messages

Identify open-source functions: BinDiff



IDA - fcClientDllv6.dll2.idb (fcClientDll.dll) C:¥Users¥11332¥Desktop¥fcClientDllv6.dll2.idb
File Edit Jump Search View Debugger Lumina Options Windows Help
Local Windows debugger
Library function
Regular function
Instruction
Data
Unexplored
External symbol
Lumina function
Functions
Matched Functions

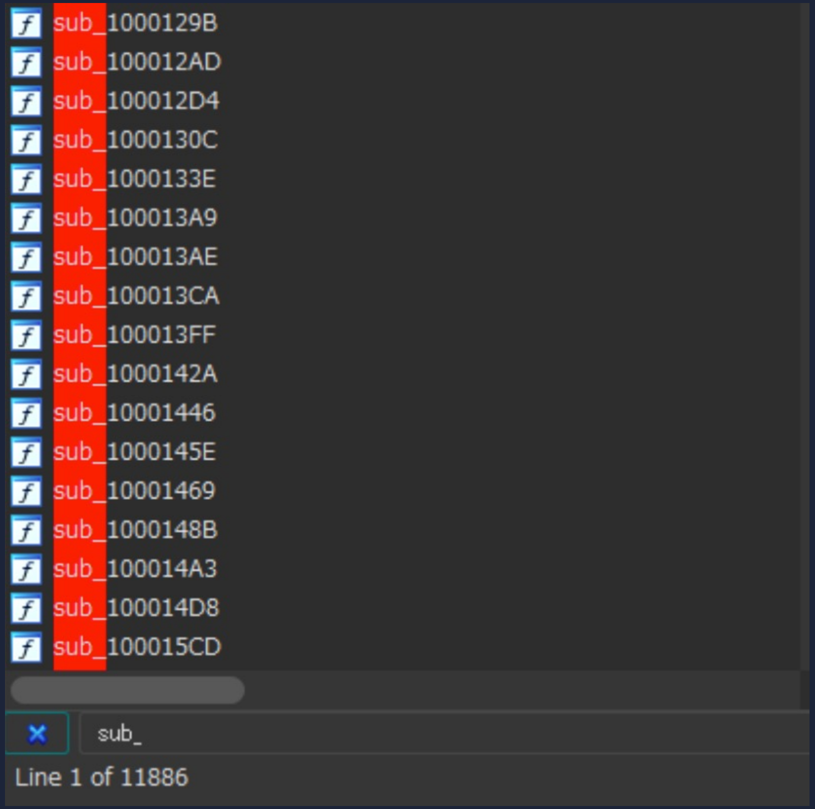
Function name
boost__asio__error___dynami
boost__asio__error___dynami
boost__asio__error___dynami
boost__asio__error___dynami
boost__asio__error___dynami
boost__asio__error___dynami
boost__asio__error___dynami
boost__asio__error___dynami
boost__asio__error___dynami
boost__asio__error___dynami
boost__asio__error___dynami
boost__asio__error___dynami
boost__asio__error___dynami
boost__asio__error___dynami
boost__asio__error___dynami
boost__asio__error___dynami
boost__asio__error___dynami
boost__asio__error___dynami
boost__asio__error___dynami
boost__asio__error___dynami
boost__asio__error___dynami

```
e::protobuf::UnescapeCEscapeSequences(_BYTE *a1, _BYTE *a2)

p+0h] [ebp-188h]
+40h] [ebp-148h]
 [esp+44h] [ebp-144h] BYREF
 [esp+70h] [ebp-118h] BYREF
 [esp+8Ch] [ebp-FCh] BYREF
 [esp+B8h] [ebp-D0h] BYREF
 [esp+E4h] [ebp-A4h] BYREF
/ [esp+110h] [ebp-78h] BYREF
1; // [esp+168h] [ebp-20h]
[esp+16Ch] [ebp-1Ch]
sp+173h] [ebp-15h]
[esp+174h] [ebp-14h]
[esp+178h] [ebp-10h]
p+184h] [ebp-4h]




v14 && *v15 && *v15 != 92 )







92 )

+v15 )

```

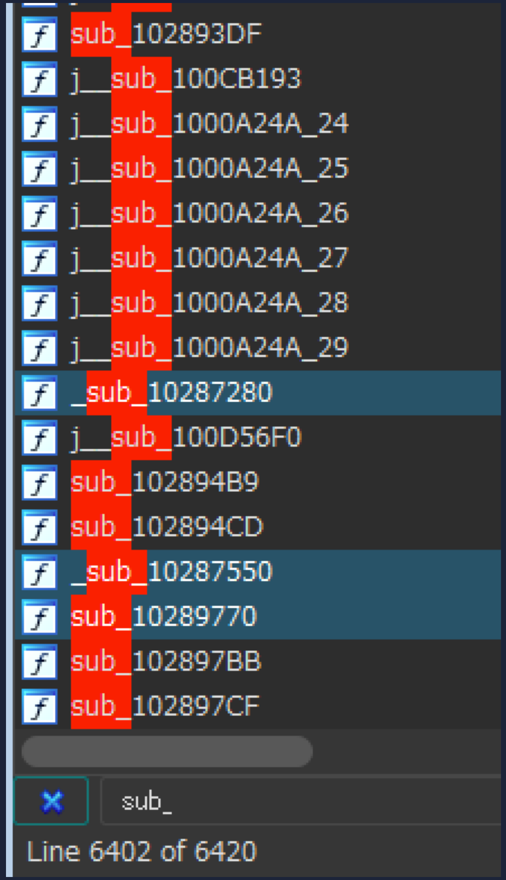
Similarity	Confider	Change	EA Primary	Name Primary	EA Secondan	Name Secondary
0.90	0.99	G---E-C	101A6350	google:		e::protobuf::UnescapeCEscapeSequences(char cor
0.00	0.02	GI--EL-	1000573F	aa_to_		e::protobuf::UnescapeCEscapeSequences(char cor

We can import symbols from matched library functions

FLIRT+ Source code review + BinDiff Result



```
sub_1000129B
sub_100012AD
sub_100012D4
sub_1000130C
sub_1000133E
sub_100013A9
sub_100013AE
sub_100013CA
sub_100013FF
sub_1000142A
sub_10001446
sub_1000145E
sub_10001469
sub_1000148B
sub_100014A3
sub_100014D8
sub_100015CD

sub_
Line 1 of 11886
```

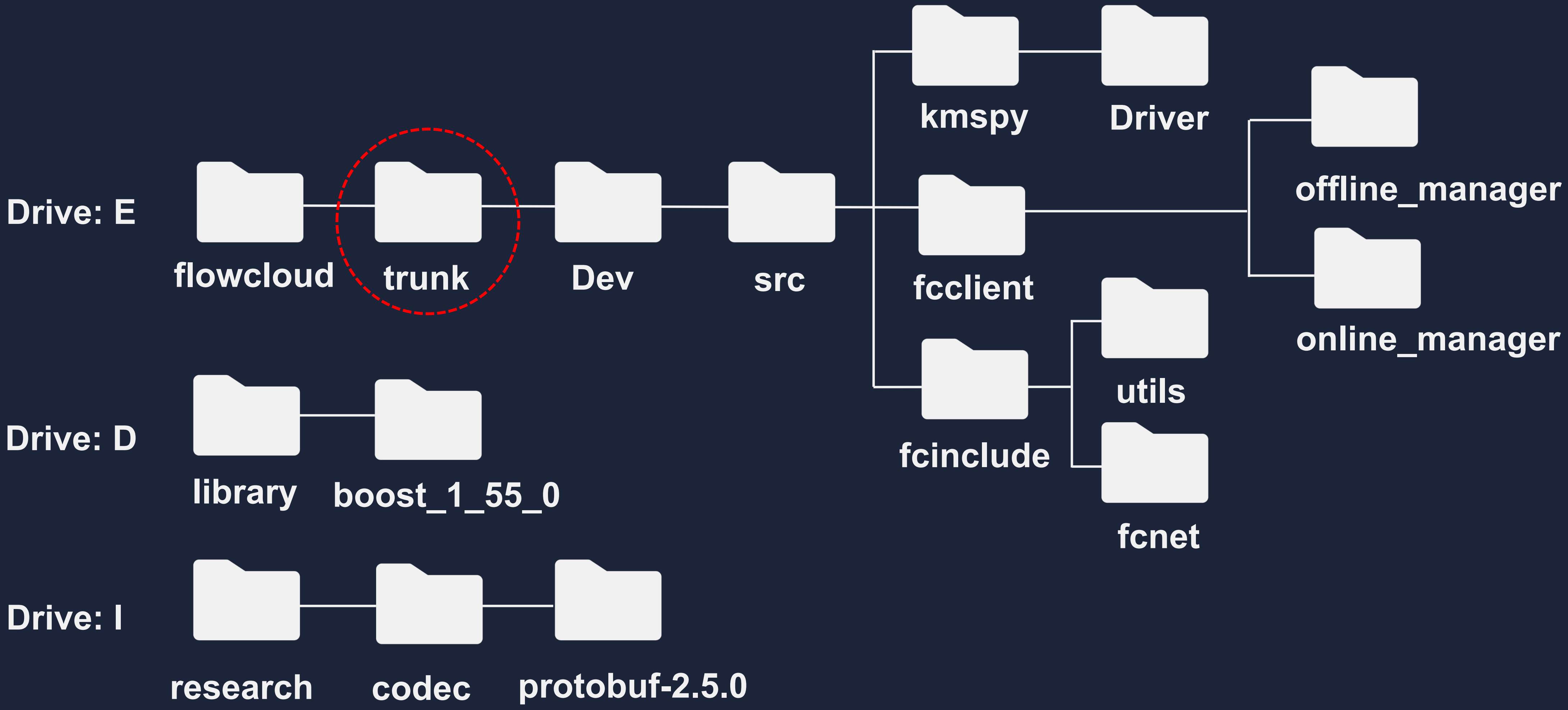


```
sub_102893DF
j__sub_100CB193
j__sub_1000A24A_24
j__sub_1000A24A_25
j__sub_1000A24A_26
j__sub_1000A24A_27
j__sub_1000A24A_28
j__sub_1000A24A_29
_sub_10287280
j__sub_100D56F0
sub_102894B9
sub_102894CD
_sub_10287550
sub_10289770
sub_102897BB
sub_102897CF

sub_
Line 6402 of 6420
```

Combination of IDA FLIRT & manual source review & Bindiff
Not 100% accurate but we could identify about 5000 library functions.

Developer Profiling

FlowCloud Development structure



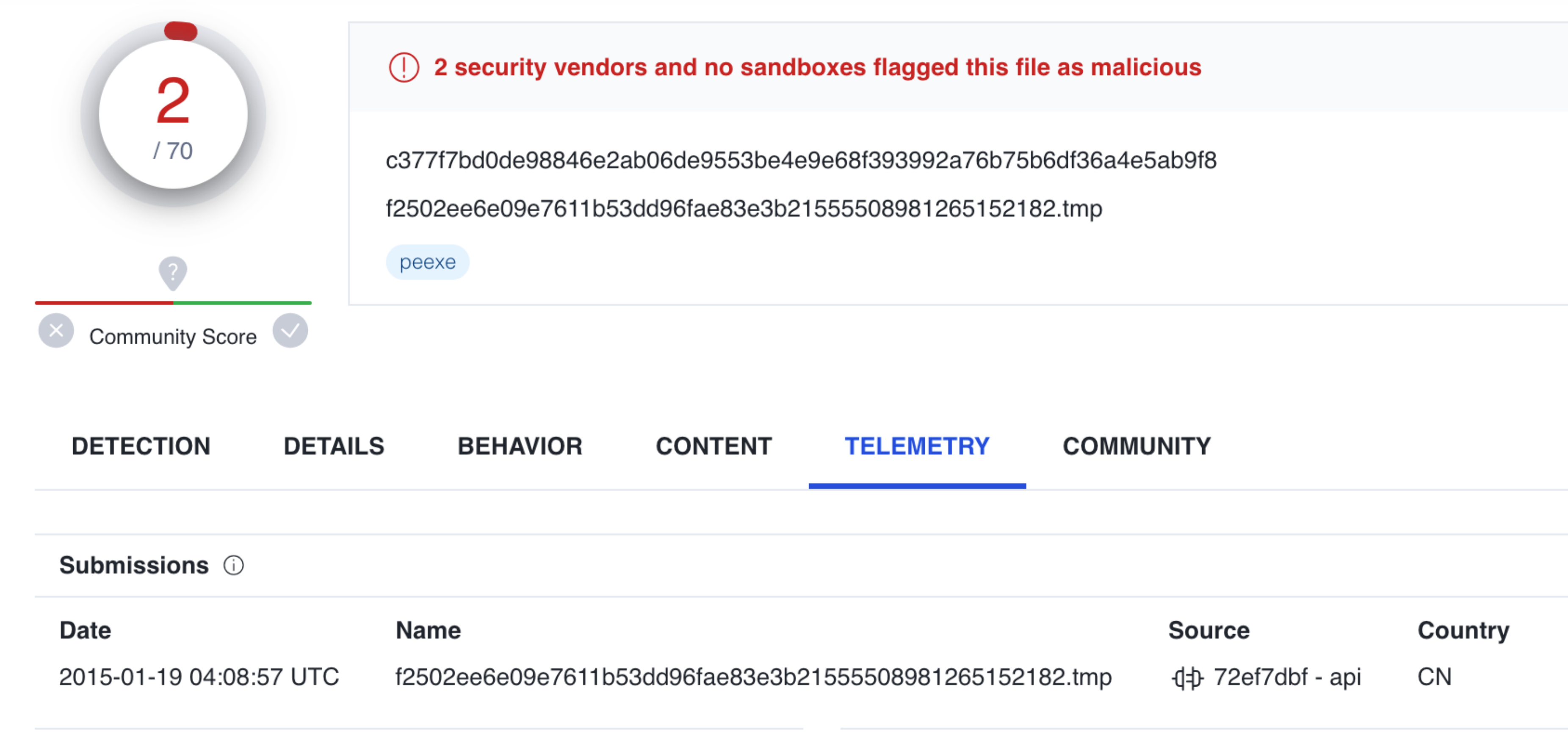
The developer(s) use SVB for source codes management

FlowCloud Open-source Library Components

- MSVC 9.0 (Visual Studio 2008 SP1)
- Protocol Buffers 2.5.0 : **Released 2015-03-25**
- ZThread 2.3.x (Probably 2.3.2) : **Release 2005-03-13**
- boost 1.55.0 : **Release 2013-11-11**
- SQLite 3.7.16 : **Release 2013-03-18**

Compiler & Open Sources versions are old, FlowCloud was developed first around 2015?

Back to 2015: FlowCloud Uninstaller



2 security vendors and no sandboxes flagged this file as malicious

c377f7bd0de98846e2ab06de9553be4e9e68f393992a76b75b6df36a4e5ab9f8

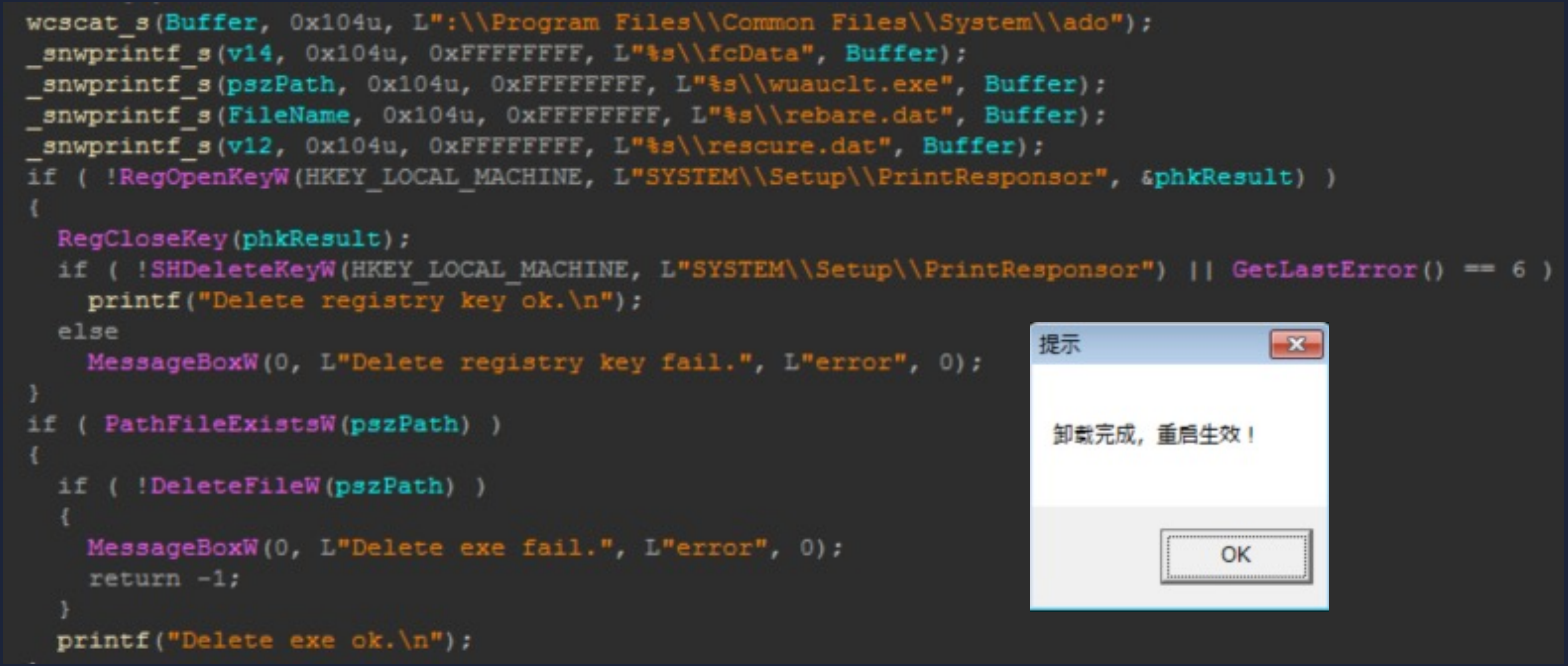
f2502ee6e09e7611b53dd96fae83e3b21555508981265152182.tmp

Submissions

Date	Name	Source	Country
2015-01-19 04:08:57 UTC	f2502ee6e09e7611b53dd96fae83e3b21555508981265152182.tmp	72ef7dbf - api	CN

h:¥work¥FlowCloud¥trunk¥Dev¥src¥fcClient¥Release¥uninstall.pdb

Back to 2015: FlowCloud Uninstaller



```
wcscat_s(Buffer, 0x104u, L":\\Program Files\\Common Files\\System\\ado");
_snwprintf_s(v14, 0x104u, 0xFFFFFFFF, L"%s\\fcData", Buffer);
_snwprintf_s(pszPath, 0x104u, 0xFFFFFFFF, L"%s\\wuauclt.exe", Buffer);
_snwprintf_s(FileName, 0x104u, 0xFFFFFFFF, L"%s\\rebare.dat", Buffer);
_snwprintf_s(v12, 0x104u, 0xFFFFFFFF, L"%s\\rescure.dat", Buffer);
if ( !RegOpenKeyW(HKEY_LOCAL_MACHINE, L"SYSTEM\\Setup\\PrintResponsor", &phkResult) )
{
  RegCloseKey(phkResult);
  if ( !SHDeleteKeyW(HKEY_LOCAL_MACHINE, L"SYSTEM\\Setup\\PrintResponsor") || GetLastError() == 6 )
    printf("Delete registry key ok.\n");
  else
    MessageBoxW(0, L"Delete registry key fail.", L"error", 0);
}
if ( PathFileExistsW(pszPath) )
{
  if ( !DeleteFileW(pszPath) )
  {
    MessageBoxW(0, L"Delete exe fail.", L"error", 0);
    return -1;
  }
  printf("Delete exe ok.\n");
```

卸载完成，重启生效！

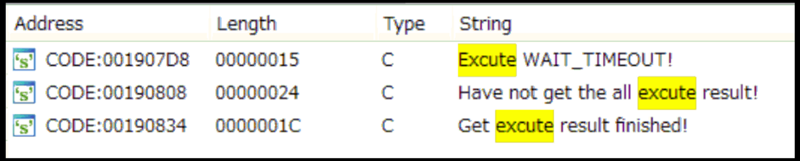
We could confirm FlowCloud existed in 2015 (at least)

Why old RAT is still effective?

- Loading decrypted payload in memory still works for circumventing

3rd RAT: Excute

- Used in Lateral Movement phase



Address	Length	Type	String
CODE:001907D8	00000015	C	Excute WAIT_TIMEOUT!
CODE:00190808	00000024	C	Have not get the all excute result!
CODE:00190834	0000001C	C	Get excute result finished!

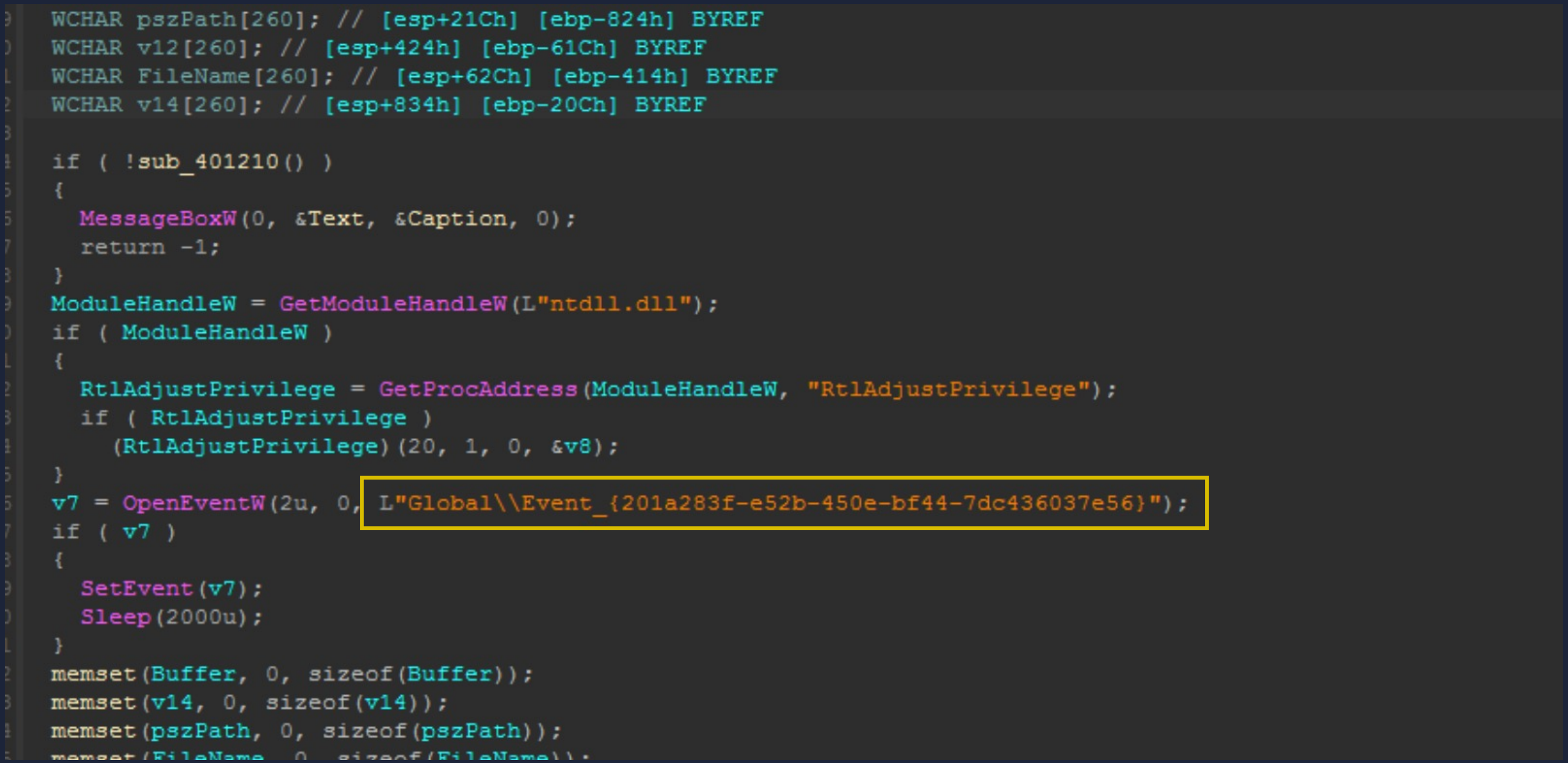
- 2 Types
  - DLL (DLL Side-Loading) or EXE
  - RAT(Excute) is decrypted and executed on memory

```
C:¥Program Files (x86)¥Microsoft SDKs¥Windows¥v7.1A¥Lib¥msicuu2.exe
C:¥Program Files (x86)¥Microsoft SDKs¥Windows¥v7.1A¥Lib¥mscoree.dll
```

```
C:¥Program Files (x86)¥Common Files¥Java¥Java Update¥juscheck.exe
```

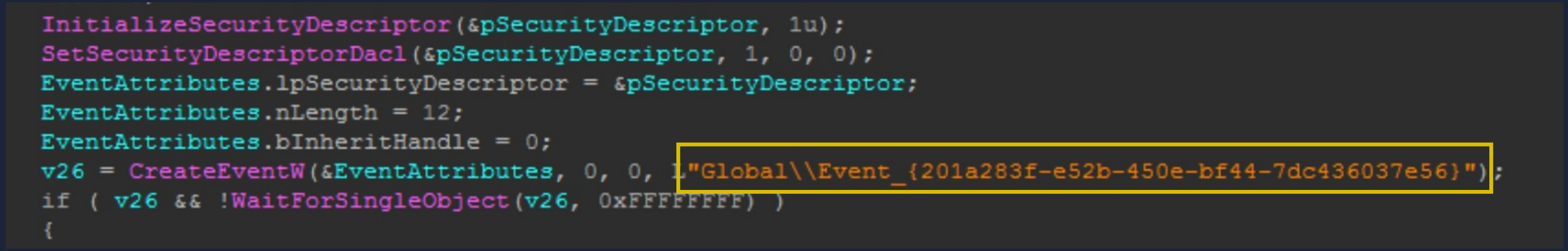
Tick used "Excute RAT". (2008 - 2019)

A string is persistent until now



```
WCHAR pszPath[260]; // [esp+21Ch] [ebp-824h] BYREF
WCHAR v12[260]; // [esp+424h] [ebp-61Ch] BYREF
WCHAR FileName[260]; // [esp+62Ch] [ebp-414h] BYREF
WCHAR v14[260]; // [esp+834h] [ebp-20Ch] BYREF

if ( !sub_401210() )
{
  MessageBoxW(0, &Text, &Caption, 0);
  return -1;
}
ModuleHandleW = GetModuleHandleW(L"ntdll.dll");
if ( ModuleHandleW )
{
  RtlAdjustPrivilege = GetProcAddress(ModuleHandleW, "RtlAdjustPrivilege");
  if ( RtlAdjustPrivilege )
    (RtlAdjustPrivilege)(20, 1, 0, &v8);
}
v7 = OpenEventW(2u, 0, L"Global\\Event_{201a283f-e52b-450e-bf44-7dc436037e56}");
if ( v7 )
{
  SetEvent(v7);
  Sleep(2000u);
}
memset(Buffer, 0, sizeof(Buffer));
memset(v14, 0, sizeof(v14));
memset(pszPath, 0, sizeof(pszPath));
memset(FileName, 0, sizeof(FileName));
```

2015
Uninstall.exe



```
InitializeSecurityDescriptor(&pSecurityDescriptor, 1u);
SetSecurityDescriptorDacl(&pSecurityDescriptor, 1, 0, 0);
EventAttributes.lpSecurityDescriptor = &pSecurityDescriptor;
EventAttributes.nLength = 12;
EventAttributes.bInheritHandle = 0;
v26 = CreateEventW(&EventAttributes, 0, 0, L"Global\\Event_{201a283f-e52b-450e-bf44-7dc436037e56}");
if ( v26 && !WaitForSingleObject(v26, 0xFFFFFFFF) )
{
```

2023
fcClientDll.dll

Conclusion

Takeaways

- USB is a classic technique, however still aggressively used.
  - Device Control is a basic counter measure
- Compiler & Open-source library versions can be useful for research & hunting
- We could uncover. FlowCloud already existed in 2015
  - 4 years before public information
- Memory region is still sweet spot for adversaries
  - Memory scan approach is effective for defenders

Questions?

https://github.com/0xebfehat/2023_flowcloud

@8th_grey_owl

Co.Tomorrowing

MACNICA